

Jarkko Heikkinen

## **LANGATON MITTAUSJÄRJESTELMÄ**

Opinnäytetyö  
Kajaanin ammattikorkeakoulu  
Tekniikan ala  
Tietotekniikan koulutusohjelma  
Syksy 2014



Koulutusala Tekniikan ala	Koulutusohjelma Tietotekniikan koulutusohjelma
Tekijä(t) Jarkko Heikkinen	
Työn nimi Langaton mittausjärjestelmä	
Vaihtoehtoiset ammattiopinnot	Toimeksiantaja Markku Karppinen, Kajaanin ammattikorkeakoulu
Aika Syksy 2014	Sivumäärä ja liitteet 53
<p>Tässä insinööriytyössä oli tarkoituksena toteuttaa Kajaanin ammattikorkeakoululle langaton mittausjärjestelmä, jota voitaisiin jatkossa hyödyntää opetuskäytössä. Järjestelmän tuli kyetä mittaamaan lämpötilaa, ilmankosteutta sekä ilmanpainetta, ja mittausdatan lähetys piti tapahtua jatkuvana datavirtana langattomasti Bluetoothin avulla. Järjestelmä tuli kasata tarvittavineen komponentteineen ja laitteineen, sekä toteuttaa C-kielellä järjestelmälle ohjelmakoodi. Järjestelmälle piti myös suunnitella ja tehdä käyttöliittymäsovellus.</p> <p>Työn teoriaosuudessa käsitellään työssä pääsääntöisesti olevia ja esiintyviä asioita ja teknologioita, kuten Raspberry Pi:tä, sekä kahta eri tiedonsiirtomenetelmää: I2C-väylää sekä Bluetoothia. Teoriaosuudessa tuodaan myös esiin, kuinka järjestelmän kehittämiseen tarkoitettu ympäristö saadaan käyttöön tarvittavineen asetuksineen, sekä järjestelmän fyysiset kytkennät ja ohjelmakoodin toteutus pääpiirteittäin.</p> <p>Työn käytännön osuus koostuu lähinnä järjestelmälle toteutettavasta C-kielisestä ohjelmakoodista, joka kehitettiin Eclipsen ja Qt Creatorin avulla. Antureilta lukeminen sekä Bluetooth-tiedonsiirto toteutettiin kannettavalla tietokoneella Eclipsen avulla käyttäen hyväksi ristiinkääntäjää. Qt Creatorilla tehtiin järjestelmälle käyttöliittymä suoraan Raspberry Pi:llä.</p> <p>Elektronikan osalta järjestelmä koostuu kahdesta Raspberry Pi -minitietokoneesta, kahdesta anturista sekä pienestä, Raspberry Pi:n GPIO-porttiin liitettävästä LCD-näytöstä. Toiseen Raspberry Pi:hin on liitetty kaksi eri anturia: lämpötila- ja ilmankosteusanturi sekä paineanturi. Mittausdata lähetetään langattomasti Bluetoothin avulla toiselle Raspberry Pi:lle, johon on kytketty pieni LCD-näyttö. Mittaustulokset saadaan näkyviin järjestelmälle suunnitellun käyttöliittymäsovelluksen avulla.</p> <p>Työn lopputuloksena saatiin lähes toimiva mittausjärjestelmä. Järjestelmä mittaa lämpötilaa sekä ilmankosteutta ja lähettää mittausdatan jatkuvana datavirtana eteenpäin Bluetoothin avulla. Paineenmittaus toimi hetken, ja käyttöliittymäsovellusta ei saatu lopulta toimimaan.</p>	
Kieli	Suomi
Asiasanat	Raspberry Pi, I2C, Bluetooth, C-kieli, Langaton
Säilytyspaikka	<input type="checkbox"/> Verkkokirjasto Theseus <input type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto

School Engineering	Degree Programme Information Technology
Author(s) Jarkko Heikkinen	
Title Wireless measurement system	
Optional Professional Studies	Commissioned by Markku Karppinen, Kajaani university of applied sciences
Date Autumn 2014	Total Number of Pages and Appendices 53
<p>The aim of this thesis was to implement a wireless measurement system which could be utilized in teaching in the future. The task was commissioned by Kajaani University of Applied Sciences. The system had to be able to measure temperature, humidity and air pressure. The data was to be transferred wirelessly by Bluetooth and it had to be continuous. The system was to be assembled from its necessary components and devices, as well as implement the program code with C language. The user interface application for the system had to be designed and created as well.</p> <p>The theoretical part of this thesis is mainly focused on issues and technologies appearing in the thesis. For example, Raspberry Pi and two different data transfer methods: I2C bus and Bluetooth. The theory part also introduces how the environment for the systems development is put to use with the needed settings, as well as physical connections of the system and the implementation of the program code in general.</p> <p>The practical part of the thesis consists mainly of the program code which was implemented for the system. The development of the program code was made by Eclipse and Qt Creator. Sensor reading and Bluetooth transfer was implemented with a laptop computer and Eclipse using a cross compiler. The user interface was made with Qt Creator straight on the Raspberry Pi.</p> <p>The electronic part consists of two Raspberry Pi mini computers, two sensors and a little LCD screen which can be attached to the Raspberry Pi's GPIO port. The temperature and humidity sensor, as well as pressure sensor is attached to another Raspberry Pi. The measurement data is transferred wirelessly by Bluetooth to another Raspberry Pi, which has a small LCD screen attached to it. The measurement data can be seen with the user interface application designed for the system.</p> <p>The final result is an almost working system which measures temperature and humidity and sends the measurement data forward continuously by Bluetooth. The pressure measurement worked awhile and wasn't able to get the user interface application working in the end.</p>	
Language of Thesis	Finnish
Keywords	Raspberry Pi, I2C, Bluetooth, C-language, Wireless
Deposited at	<input type="checkbox"/> Electronic library Theseus <input type="checkbox"/> Library of Kajaani University of Applied Sciences

## ALKUSANAT

Haluan kiittää Kajaanin ammattikorkeakoulun Markku Karppista työni mahdollistamisesta. Haluan kiittää myös työni valvojaa Tuomo Rantalaä työn ohjauksesta sekä Eero Soinista kielellisestä ohjauksesta. Kiitokset myös Eero Huuskolle sekä kaikille muille työssä auttaneille.

# SISÄLLYS

1 JOHDANTO	1
2 RASPBERRY PI	2
2.1 Raspberry Pin syntyminen	2
2.2 Raspberry Pi:n laitteisto-ominaisuudet	3
2.3 Raspbian	6
3 I2C-VÄYLÄ	7
3.1 Versiot	7
3.2 Väylän rakenne	8
3.3 Viestiprotokolla	9
3.4 Kommunikointi	9
4 BLUETOOTH	11
4.1 Versiot	11
4.2 Arkkitehtuuri	12
4.2.1 Protokollapino	13
4.2.2 L2CAP-protokollakerros	15
4.3 Fyysinen kerros	16
4.3.1 Taajuusmodulointi	16
4.3.2 Taajuushyppely	17
4.4 Viestikehykset	19
4.5 Laiteosoitteet	21
5 KEHITYSYMPÄRISTÖN KÄYTTÖÖNOTTO	23
5.1 SSH-yhteyden luominen	23
5.2 Ristiinkääntäjän asennus	25
5.3 Bcm2835-kirjaston asentaminen Raspberry Pi:lle	27
5.4 Eclipse-kehitysympäristö	27
5.4.1 Eclipsen asennus	28
5.4.2 Ristiinkääntäjän käyttöönotto	28
5.4.3 Bcm2835-kirjaston käyttöönotto Eclipse-projektissa	29
5.4.4 SSH-yhteys Raspberry Pi:lle	30
5.5 Qt Creator-kehitysympäristö	34

6 TYÖN SUORITUS	35
6.1 Lämpötilan ja kosteuden mittaus	36
6.1.1 Anturin kytkentä	37
6.1.2 Lämpötilan ja kosteuden lukeminen anturilta	38
6.2 Ilmanpaineen mittaus	39
6.2.1 Anturin kytkentä	39
6.2.2 Ilmanpaineen lukeminen anturilta	40
6.3 Bluetooth-yhteys	42
6.3.1 Bluetooth USB-adapterin käyttöönotto Raspberry Pi:ssä	42
6.3.2 Bluetooth-tiedonsiirto	43
6.4 LCD-näyttö	45
6.4.1 LCD-näytön kytkeminen	45
6.4.2 LCD-näytön alustaminen C-ohjelmointikielellä	45
6.5 Käyttöliittymäsovellus	46
7 TYÖN/TULOSTEN ANALYSOINTI	49
8 YHTEENVETO	51
LÄHTEET	52

## SYMBOLILUETTELO

A/D	Analog to Digital
ABI	Application Binary Interface
ACK	Acknowledgement
ACL	Asynchronous Connection-Less
AFH	Adaptive Frequency Hopping
AMP	Alternative MAC/PHY
ARM	Advanced RISC Machines
BLE	Bluetooth Low Energy
CPU	Central Processing Unit
EDR	Enhanced Data Rate
FEC	Forward Error Correction
FFC	Flat Flex Cable
GCC	GNU Compiler Collection
GPIO	General Purpose Input/Output
GPU	Graphics Processing Unit
HCI	Host Controller Interface
HD	High Definition
HDMI	High Definition Multimedia Interface
HS	High Speed
I2C	Inter-Integrated Circuit
IC	Integrated Circuit
IP	Internet Protocol
ISM	Industrial, Scientific and Medical
L2CAP	Logical Link Control and Adaptation Protocol
LAP	Lower Address Part
LCD	Liquid-Crystal Display
LED	Light-Emitting Diode
LMP	Link Manager Protocol
LSB	Least Significant Bit
MAC	Media Access Control
MSB	Most Significant Bit
NAP	Non-significant Address Part

OSI	Open Systems Interconnection
PSM	Protocol and Service Multiplexor
QML	Qt Meta Language
RFCOMM	Radio Frequency Communication
SCL	Serial Clock
SCO	Synchronous Connection Oriented
SD	Secure Digital
SDA	Serial Data
SDRAM	Synchronous Dynamic Random-Access Memory
SoC	System on Chip
SPI	Serial Peripheral Interface
SSH	Secure Shell
SSP	Secure Simple Pairing
TFT	Thin-Film-Transistor
UAP	Upper Address Part
UART	Universal Asynchronous Receiver Transmitter
USB	Universal Serial Bus
USCL	Serial Clock
USDA	Serial Data
WLAN	Wireless Local Area Network
WPAN	Wireless Personal Area Network



## 1 JOHDANTO

Insinööriyön aihetta miettiessäni olin tehnyt sen päätöksen, että työn tulisi sisältää paljon ohjelmointia, koska mielestäni ohjelmointitaitoni olivat vielä täysin riittämättömät. Siksi en halunnutkaan kysyä insinööriyölleni aihetta yrityksiltä, koska niiden tarjoamat aiheet eivät olisi välttämättä täyttäneet insinööriyöni aiheelle asettamiani tavoitteita. Tästä syystä otin avukseni Kajaanin ammattikorkeakoulun projekti-insinööri Markku Karppisen, jonka kanssa rupesimme suunnittelemaan minulle insinööriyön aihetta.

Insinööriyön aiheekseni tuli lopulta langaton mittausjärjestelmä, johon kuitenkin heti työn alkumetreillä tuli alkuperäisiin suunnitelmiin hieman muutoksia koskien antureita lukevaa alustaa. Alun perin antureita oli tarkoitus lukea Atmelin ATtiny-mikrokontrollerilla, mutta lukuisien eri ongelmien ilmaantuessa muun muassa Atmelin AVR Studio -kehitysympäristön kanssa päätimme vaihtaa AVR:n kontrollerin Raspberry Pi:hin. Alun perin oli tarkoitus myös toteuttaa mittausjärjestelmälle Android-sovellus, mutta lopulta sovellukselle ei ollutkaan enää tarvetta.

Lopullinen järjestelmä koostuu kahdesta Raspberry Pi -minitietokoneesta, kahdesta anturista ja pienestä LCD-näytöstä (Liquid-Crystal Display). Toinen Raspberry Pi lukee antureita ja lähettää mittausdatan langattomasti Bluetoothin avulla toiselle Raspberry Pi:lle, johon kytketystä LCD-näytöstä on mahdollista tarkkailla mittaustuloksia.

Tehtävänä oli toteuttaa Kajaanin ammattikorkeakoululle langaton mittausjärjestelmä, jolla mitataan ilman lämpötilaa, kosteutta sekä painetta. Mittausdata tuli siirtää jatkuvana datavirtana langattomasti Bluetoothin avulla toiselle Raspberry Pi:lle, johon kytkettyyn LCD-näyttöön mittaustulokset saadaan näkyviin järjestelmälle toteutettavan käyttöliittymän avulla. Mittausjärjestelmää on tarkoitus hyödyntää jatkossa opetuskäytössä.

## 2 RASPBERRY PI

Raspberry Pi on edullinen, luottokortin kokoinen minitietokone, jota käytetään tavallisella näppäimistöllä ja hiirellä. Se on mahdollista kytkeä joko tietokoneen näyttöön tai televisioon. Raspberry Pi:llä voi muun muassa toteuttaa erilaisia elektroniikan projekteja, ja se mahdollistaa kaikenikäisten ihmisten tutustua tietojenkäsittelyyn sekä opettaa ohjelmoimaan eri ohjelmointikielillä, kuten Pythonilla tai C-kielillä. Raspberry Pi:llä voi tehdä oikeastaan kaikkea mitä normaalillakin tietokoneella: selata Internetiä, toistaa HD-tasoisista (High-Definition) videoita tai pelata pelejä. [1.]

### 2.1 Raspberry Pin syntyminen

Raspberry Pin kehitys lähti liikkeelle vuonna 2006, kun Cambridgen yliopiston Eben Upton, Rob Mullins, Jack Lang ja Alan Mycroft huolestuivat tietojenkäsittelytieteiden koulutusohjelmaan hakevien koko ajan laskussa olevista numeroista ja taidoista. 1990-luvulla hakijat olivat kokeneita ohjelmoijia, kun taas 2000-luvulla hakijoilla ei ollut juuri minkäänlaista kokemusta ohjelmoinnista. Nykyajan tietokoneet ovat kalliita, ja niillä ohjelmointi ei ole enää niin yksinkertaista kuin ennen oli esimerkiksi Amigalla tai Commodore 64:llä, joten vuosina 2006-2008 kehiteltiin ensimmäiset prototyypit helpottamaan ohjelmointiin tutustumista. Nii-tä kutsutaan nykyään nimellä Raspberry Pi. [2.]

Vuoteen 2008 mennessä mobiililaitteiden prosessorit olivat jo niin kehittyneitä, että niillä voitiin tuottaa laadukasta multimediaa. Tällä ominaisuudella oletettiin olevan positiivinen vaikutus alustan kehityksessä myös niihin, joilla ei ollut aluksi kiinnostusta täysin ohjelmointiin tarkoitettulla alustalla. Eben Uptonin, Rob Mullinsin, Jack Langin ja Alan Mycroftin mukaan tulivat vielä Pete Lomas ja David Braben, jotka perustivat Raspberry Pi Foundationin. Kolmen vuoden kuluttua alkoi Raspberry Pi Model B:n massatuotanto, ja kahdessa vuodessa sitä on ostettu yli kaksi miljoonaa kappaletta. [2.]

## 2.2 Raspberry Pi:n laitteisto-ominaisuudet

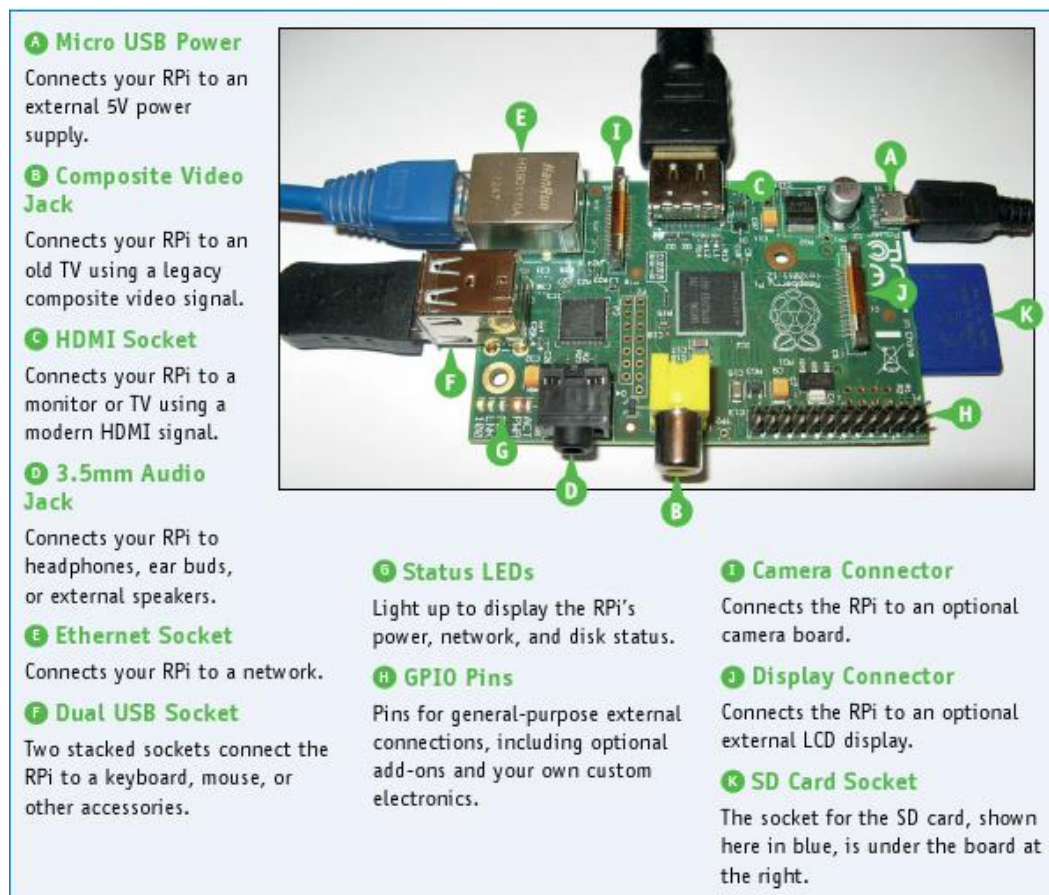
Raspberry Pi:stä on kaksi erilaista mallia - malli A ja malli B. Niissä on hieman poikkeavia ominaisuuksia. Malli B on näistä kahdesta kehittyneempi. Siinä on malli A:n ominaisuuksien lisäksi kaksi kertaa enemmän keskusmuistia, yksi USB-portti (Universal Serial Bus) enemmän sekä 10/100 Mbit/s Ethernet-portti, joka malli A:sta puuttuu kokonaan. Malli A on kuitenkin mahdollista yhdistää verkkoon esimerkiksi USB-porttiin liitettävän WLAN-adapterin (Wireless Local Area Network) avulla. Koska malli A sisältää vähemmän ominaisuuksia kuin malli B, sen virrankulutus on hieman pienempi sekä myyntihinta on myös hieman edullisempi kuin malli B:n. Mallien ominaisuudet on nähtävissä taulukossa 1. [4.]

Taulukko 1. Raspberry Pi -mallien ominaisuudet [3] [4]

	Model A	Model B
<b>Chip</b>	Broadcom BCM2835 SoC	
<b>CPU</b>	700 MHz ARM1176JZF-S core(ARM11 family, ARMv6 instruction set)	
<b>GPU</b>	Broadcom Dual Core VideoCore IV @ 250 MHz	
<b>Memory</b>	256 MB SDRAM	512 MB SDRAM
<b>USB 2.0 ports</b>	1	2
<b>Video outputs</b>	HDMI (rev 1.3 & 1.4), Composite RCA (PAL and NTSC)	
<b>Audio outputs</b>	3.5mm jack, HDMI	
<b>Onboard storage</b>	SD/MMC/SDIO card slot	
<b>Onboard network</b>	None	10/100 Mbit/s Ethernet (8P8C), RJ45 jack
<b>Power ratings</b>	300 mA (1.5 W)	700 mA (3.5W)
<b>Power source</b>	5 V via MicroUSB or GPIO header	
<b>Size</b>	8,6 cm x 5,4 cm x 1,5 cm (45g)	8,6 cm x 5,4 cm x 1,7 cm (45g)
<b>Operating systems</b>	Linux	

Raspberry Pi:n käyttäminen vaatii viisi asiaa: SD-kortin (Secure Digital), jossa on käyttöjärjestelmä, käyttöjännitteen kortille, näytön sekä hiiren ja näppäimistön. Raspberry Pi vaatii käynnistyäkseen 5 V käyttöjännitteen. Käyttöjännitteen voi syöttää kortille joko suoraan USB-kaapelin kautta esimerkiksi tietokoneen USB-portista tai verkkopistokkeesta erillisen jännitemuuntimen kautta, joka muuntaa 230 V vaihtojännitteen 5 V tasajännitteeksi. Yksi mahdollisuus on myös syöttää 5 V Raspberry Pi:n 5 V:n GPIO-pinniin (General Purpose Input/Output). Kuva Raspberry Pi:ltä saadaan joko tietokoneen näytölle tai televisioon HDMI-kaapelin (High Definition Multimedia Interface) avulla. USB-portteihin voidaan kytkeä näppäimistö ja hiiri. Periaatteessa Raspberry Pi:n käyttäminen onnistuu pelkästään näppäimistöllä, jos ei käytetä ollenkaan graafista käyttöliittymää, vaan komennot ajetaan Linux-käyttöjärjestelmälle tyypilliseen tapaan suoraan komentoriville.

Muita liityntöjä Raspberry Pi:ssä ovat muun muassa komposiittivideo, jolla on mahdollista siirtää kuvaa ja 3.5 mm audio-liitin, johon on mahdollista kytkeä kuulokkeet tai kaiuttimet. Kortilta löytyy myös RJ-45-mallinen Ethernet-liitin, jos Raspberry Pi:n haluaa yhdistää verkkoon. Liittynät ovat nähtävissä kuvassa 1.



Kuva 1. Raspberry Pi, malli B:n liitynnät ja komponentit [7]

GPIO-pinnit ovat pinnejä, joita on mahdollista ohjata itse kirjoitetun ohjelman avulla. Raspberry Pi:stä löytyy 2x13-pinninen liitin, joista GPIO-pinnejä on kahdeksan kappaletta sekä kaksi pinniä I2C:lle (Inter-Integrated Circuit), viisi pinniä SPI:lle (Serial Peripheral Interface) sekä kaksi pinniä UART:lle (Universal Asynchronous Receiver/Transmitter), joita kaikki voi myös silti käyttää tavallisen GPIO-pinnin tapaan. Loput pinnit ovat 3,3 V, 5 V sekä 0 V eli maa. [5.]

GPIO-pinnien jännitetaso on 3,3 V, joten tätä ei kannata ylittää, sillä Raspberry Pi:ssä ei ole ylijännitesuojaa. 5 V pinnin kautta voidaan antaa Raspberry Pi:lle sen tarvitsema käyttöjännite. Muuhun tarkoitukseen sitä ei kannata käyttää. GPIO-pinnit ja niiden toiminnot ovat nähtävissä taulukossa 2. [6.]

Taulukko 2. Raspberry Pi:n GPIO-pinnikaavio [5]

GPIO-pinninumero	Toiminto	GPIO-pinninumero	Toiminto
1	3,3 V	14	0 V (maa)
2	5 V	15	GPIO 22
3	Versio 1: GPIO 0 (SDA) Versio 2: GPIO 2 (SDA)	16	GPIO 23
4	5 V	17	3,3 V
5	Versio1: GPIO 1 (SCL) Ver- sio 2: GPIO 3 (SCL)	18	GPIO 24
6	0 V (maa)	19	GPIO 10 (MOSI)
7	GPIO 4 (GPCLK 0)	20	0 V (maa)
8	GPIO 14 (TXD)	21	GPIO 9 (MISO)
9	0 V (maa)	22	GPIO 25
10	GPIO 15 (RXD)	23	GPIO 11 (SCLK)
11	GPIO 17	24	GPIO 8 (CE0)
12	GPIO 18 (PCM_CLK)	25	0 V (maa)
13	Versio 1: GPIO 21 Versio 2: GPIO 27	26	GPIO 7 (CE1)

## 2.3 Raspbian

Raspbian on yksi Raspberry Pi:lle asennettavissa olevista ilmaisista käyttöjärjestelmistä. Sen perustajina oli Raspberry Pi:stä kiinnostunut pieni ryhmä, joka ei ole kuitenkaan millään tavalla sidoksissa Raspberry Pi Foundationiin. Raspbian perustuu Debian Wheezy - käyttöjärjestelmään, ja se on optimoitu niin, että toiminta olisi mahdollisimman sulavaa Raspberry Pi:llä. Raspbian ei ole ihan tavallinen käyttöjärjestelmä, sillä se tarjoaa yli 35000 esikäännettyä ohjelmistopakettia valmiina helposti asennettavassa muodossa. [8.]

Virallisen Debian Wheezyn armhf (ArmHardFloatPort) versio ei ole yhteensopiva Raspberry Pi:ssä käytettävän ARM-arkkitehtuurin (Advanced RISC Machines) kanssa. Näin ollen Debian Wheezyn armhf:ää on jouduttu hieman muokkaamaan, jotta sen suoritus onnistuu myös Raspberry Pi:llä. Armhf tuottaa koodia, joka hyödyntää ”hard float ABI:a”. ABI:lla (Application Binary Interface) tarkoitetaan tapaa, jolla asetetaan prosessorin rekistereitä sekä muokataan pino-tietorakennetta, ja hard float ABI asettaa liukulukuparametrit liukulukurekistereihin. [10.]

### 3 I2C-VÄYLÄ

I2C on usean isännän sarjamuotoinen tiedonsiirtoväylä, jonka Philips kehitti 80-luvun alussa. I2C-väylällä haluttiin saada aikaan mahdollisimman yksinkertainen tiedonsiirto hitaiden laitteiden välille. Väylä ei ole tarkka baudinopeudesta, sillä isäntä luo kellotaajuuden, jolla signaali tahdistetaan eikä väylä vaadi kuin kaksi johtoa tiedonsiirtoon. Väylää kutsutaankin usein nimellä kahden johdon väylä tai rajapinta. Esimerkiksi Atmel käyttää omissa mikrokontrollereissaan kahden johdon rajapinta (Two Wire Interface)-nimistä väyläratkaisua, jonka toimintaperiaate on täysin identtinen – Atmel on vain halunnut antaa sille toisen nimen, vaikka I2C ei ole rekisteröity tavaramerkki. I2C-väylän toiminta perustuu isäntä/orja-periaatteeseen. Orjilla on kaikilla oma ohjelmallinen laiteosoite, jonka perusteella isäntä antaa kullekin orjalle vuorollaan luku- tai kirjoitusoikeuden väylälle. [11.] [12.]

#### 3.1 Versiot

Alla on esitelty I2C-väylän eri versioiden ominaisuudet ja kehittyminen.

**1982:** Alkuperäinen 100 kHz:n taajuudella toimiva yksinkertainen I2C-väylä luodaan.

**1992:** Ensimmäinen standardoitu versio eli versio 1 sai alkunsa. Tähän versioon oli lisätty 400 kHz:n Fast-moodi sekä 10-bittinen osoitinmoodi, jolla väylälle mahtuvien laitteiden kapasiteetti saatiin nostettua 1008 laitteeseen.

**1998:** Versio 2:n tulo lisäsi väylään 3.4 MHz:n taajuudella toimivan High-speed-moodin sekä virransäästöominaisuuksia.

**2000:** Versio 2.1:een tehtiin pieniä parannuksia edellisestä versiosta.

**2007:** Versio 3, joka päivitettiin uudella 1 MHz:n taajuudella toimivalla Fast+-moodilla sekä laitetunnistemekanismilla.

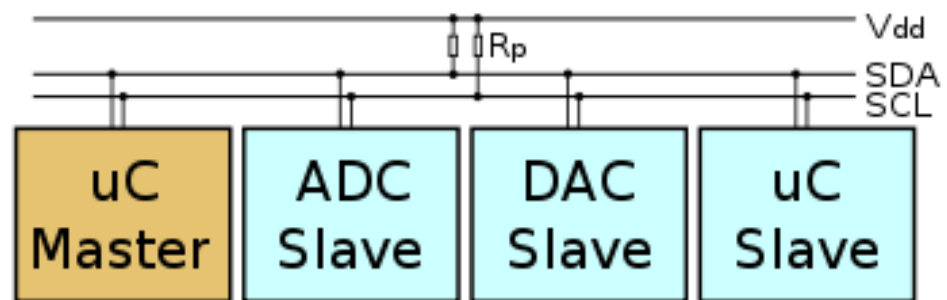
**2012:** Versio 4:n tulo lisäsi väylään uusille, ilman ylös- ja alaspäin siirtävää push-pull-logiikkaa käyttäville USDA- (Serial Data) ja USCL-linjoille (Serial Clock) entistäkin nopeamman Ultra-Fast-moodin, joka mahdollisti jopa 5 MHz:n taajuuden.

**2012:** Versio 5 korjasi joitain virheitä.

**2014:** Versio 6 on uusin standardi, johon korjattiin kaksi kuvaajaa. [12.]

### 3.2 Väylän rakenne

I2C-väylä rakentuu vain kahdesta kaksisuuntaisesta johdosta sekä kahdesta ylösvetovastuksesta. SCL (Serial Clock) on kellolinja, jolla tahdistetaan kaikki data mitä väylällä liikkuu. SDA-linjalla (Serial Data) liikkuu väylällä siirrettävä data. Nämä kaksi signaalia mahdollistavat jokaisella kellopulsilla tavun mittaisen sarjamuotoisen tiedonsiirron. I2C-väylällä isännäksi kutsutaan laitetta (mikrokontrolleri tai prosessori), joka antaa tiedonsiirtokäskyn ja ohjaa SCL-linjaa. Orjat ovat laitteita, joilla jokaisella on oma laiteosoite, jonka perusteella isäntä käskyy niitä. Väylän rakennetta on havainnollistettu kuvassa 2. [13.] [14.]



Kuva 2. I2C-väylän rakenne [15]

SCL- ja SDA-linjat kytketään jokaisesta väylällä olevasta laitteesta niitä ohjaavalle mikrokontrollerille tai prosessorille. SCL- ja SDA-linjat tulee myös kytkeä  $1,8\text{ k}\Omega$  -  $10\text{ k}\Omega$  vastuksen (yleensä n.  $4,7\text{ k}\Omega$ ) kautta  $5\text{ V}$ :n tai  $3,3\text{ V}$ :n käyttöjännitteeseen. Väylä tarvitsee vain yhden ylösvetovastuksen väylää kohden - ei siis jokaiselle väylällä olevalle laitteelle. Kummatkin linjat ovat ns. open drain -linjoja, eli kontrolleri voi asettaa lähdön nolnaan, muttei voi asettaa sitä ykköseksi, joten käyttöjännitteeseen kytketyn ylösvetovastuksen avulla linjat saadaan asetettua ykköseksi. [13.]



### 3.3 Viestiprotokolla

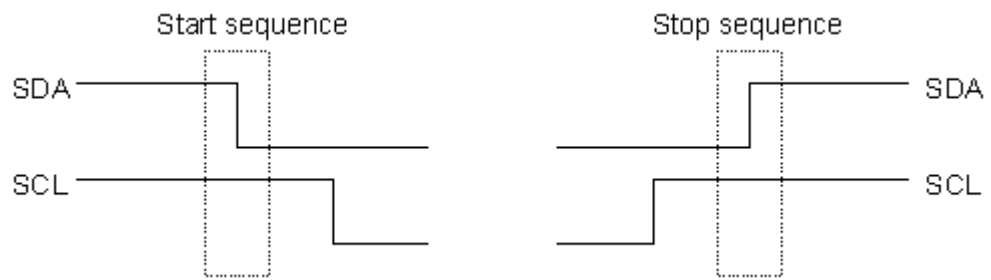
I2C-väylän viestiprotokolla käsittää kolme erilaista viestityyppiä, joista jokainen alkaa aloituskäskyllä ja päättyy lopetuskäskyyn. Nämä kolme viestityyppiä ovat: yksittäiset viestit väylään, jossa isäntä joko lukee väylästä tai kirjoittaa väylään, sekä yhdistetty viestintä, jossa isäntä tekee useita luku- tai kirjoituspyyntöjä usealle eri orjalaitteelle. [13.]

Kaikki data mikä väylällä liikkuu, lähtee aina liikkeelle isännän aloituskäskystä. Dataa siirtyy 8 bittiä kerrallaan, ja järjestys on aina MSB-bitti (Most Significant Bit) ensin. Yhdelle tiedonsiirrolle ei ole tiedonsiirto rajoja, joten jokaisessa tiedonsiirrosta voi kuljettaa niin paljon dataa kuin halutaan. [17, s. 1.]

I2C-väylä käyttää 7- ja 10-bittisiä laiteosoitteita, joista 7-bittinen on huomattavasti yleisempi. Teoriassa 7-bittisellä laiteosoitteella väylään saadaan maksimissaan 128 eri laitetta ja 10-bittisellä laiteosoitteella maksimissaan 1024 laitetta. Käytännössä ei kuitenkaan saada näin montaa laitetta kytkettyä väylään, sillä väylä varaa joitain tiettyjä osoitteita, joten esimerkiksi 7-bittisellä laiteosoitteella väylään saadaan käytännössä 112 eri laitetta. [17, s. 2.]

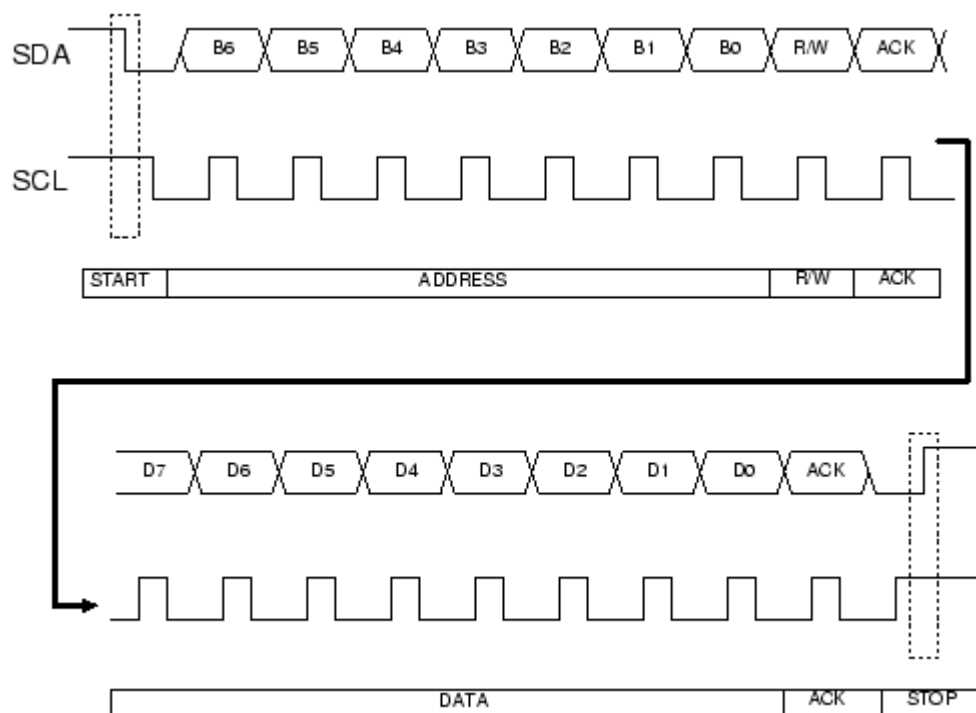
### 3.4 Kommunikointi

I2C-väylän tiedonsiirto alkaa, kun isäntä lähettää aloituskäskyn laitteelle, jonka kanssa se haluaa keskustella. Aloituskäskyssä SDA- ja SCL-linjat ovat molemmat normaalitilassa eli ykköstilassa. SDA-linja asetetaan nolaksi, kun SCL-linja on vielä ykköstilassa. Tämän jälkeen isäntä lähettää 7 bitin mittaisen (voi myös olla 10 bittiä pitkä, mutta ovat harvinaisia) laiteosoitteen, MSB-bitti ensin, orjalle, jonka kanssa se haluaa kommunikoida. Laiteosoitteen 0-bitti eli LSB-bitti (Least Significant Bit) määrittää, luetaanko laitteelta vai kirjoitetaanko laitteelle. 0-bitti on lukemista varten, ja 1-bitti on kirjoitusta varten. Kun kaikki halutut tavut on luettu tai lähetetty, isäntä lähettää orjalaitteelle lopetuskäskyn asettamalla SDA-linjan ykköstilaan, SCL-linjan ollessa ykköstilassa. Tällä saadaan annettua väylään merkki, että isäntä on valmis ottamaan liikennettä vastaan muilta laitteilta. SDA- ja SCL-linjan aloitus- ja lopetussignaali on nähtävissä kuvassa 3. [13.] [14.] [15.]



Kuva 3. I2C-väylän aloitus- ja lopetussignaalit [13]

Data siirretään SDA-linjaa pitkin tavun mittaisina jonoina MSB-bitistä alkaen. SDA-linjalle siirtyä yksi bitti dataa jokaisella SCL-pulssin nousevalla reunalla. Jokaisen siirretyn kahdeksan bitin jälkeen vastaanottava laite lähettää ACK- (Acknowledgement) eli vahvistusbitin, joten tarvitaan yhdeksän kellopulssin mittainen jakso siirtämään kahdeksan bittiä dataa. Vahvistusbitin ollessa 0 laite vastaanotti tavun ja on valmis vastaanottamaan uuden tavun. Jos vahvistusbitti on 1, laite ei ole enää valmis vastaanottamaan yhtään dataa, ja isännän tulisi tehdä lopetuskäskeä. I2C-väylän datansiirron periaate on nähtävissä kuvassa 4. [13.]



Kuva 4. I2C-väylän SDA- ja SCL-signaalit [16]

## 4 BLUETOOTH

Bluetooth on avoin lyhyen kantaman radioteknologiaan perustuva tiedonsiirtoratkaisu, joka suunniteltiin alun perin korvaamaan kaapeleita kiinteiden ja/tai liikkuvien laitteiden välillä. Se syntyi vuonna 1998, ja kehityksessä oli mukana neljä suurta tekniikan alan yritystä: Ericsson, IBM, Nokia ja Toshiba. Bluetooth oli alun perin standardoitu tiedonsiirtoratkaisu, mutta kyseisen IEEE 802.15.1-standardin ylläpito on lopetettu. [18, s. 325.] [19.]

Nykyään Bluetoothin kehittäminen jatkuu Bluetooth SIG (Special Interest Group) -nimisen ryhmittymän toimesta, johon kuuluu yhteensä yli 20000 eri tekniikan alan yritystä. [19.]

### 4.1 Versiot

Ihan ensimmäiset Bluetooth-versiot olivat 0.8 ja 0.9, ja ne olivat salaisia. Ensimmäinen ei-salainen versio julkaistiin vuonna 1999 versionimellä 1.0. Versio 1.0 sisälsi paljon virheitä, jonka seurauksena vuonna 2002 virheitä korjaamaan kehitettiin versio 1.1, jonka myös IEEE standardoi nimellä IEEE 802.15.1. Versio 1.1 päivitettiin vuonna 2003 versioon 1.2. Päivitetyn 1.2 version tärkeimpänä uutena ominaisuutena oli AFH (Adaptive Frequency Hopping) eli suomeksi taajushyppely-nimellä tunnettu tekniikka. [18, s. 325.] [19.]

Vuonna 2004 julkaistu versio 2.0 sai uutena ominaisuutena muun muassa EDR:n (Enhanced Data Rate), jonka avulla tiedonsiirtonopeus saatiin kolminkertaistettua, eli nyt tiedonsiirtonopeus oli 3 Mbit/s. 2.1-versio julkaistiin vuonna 2007, ja sen merkittävin parannus edellisestä versiosta oli SSP (Secure Simple Pairing), joka paransi Bluetoothin tietoturvaa. [19.]

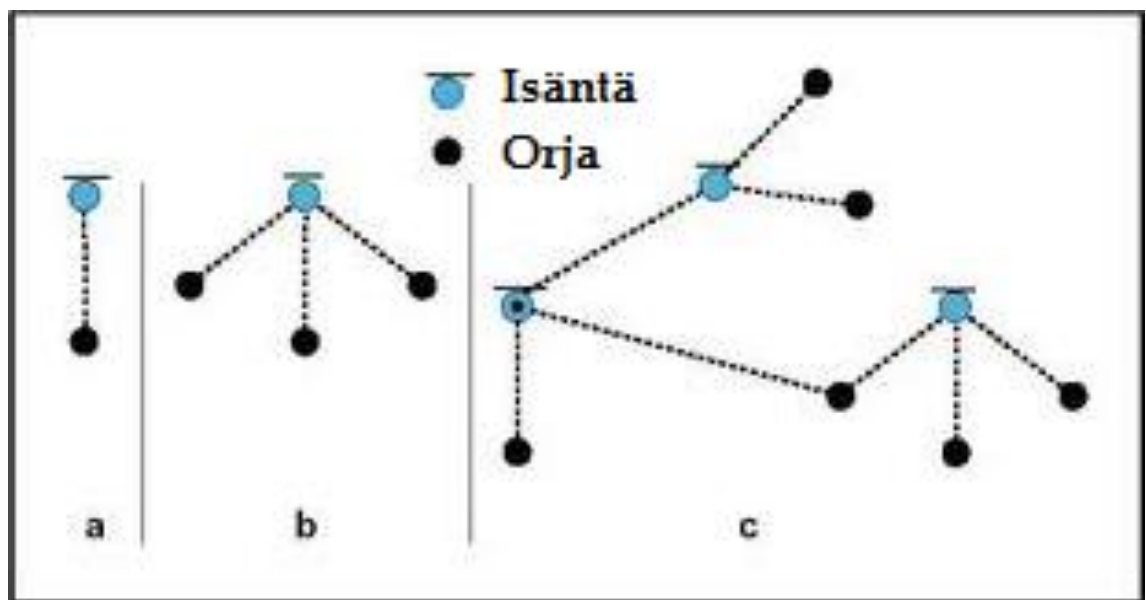
Vuonna 2009 julkaistiin versio 3.0, johon oli lisätty uudet HS- (High Speed) ja AMP (Alternative MAC/PHY)-tekniikat. High Speed -tekniikan myötä tiedonsiirtonopeus saatiin teoriassa nostettua jopa 24 Mbit/s. High Speed -tekniikka ei kuitenkaan toimi suoraan Bluetooth-linkin kautta, vaan se vaatii 802.11-linkin toimiakseen. [19.]

Vuonna 2010 julkaistu versio 4.0 sai jälleen parannusta tiedonsiirtonopeuteen. Lisäksi virrankulutusta saatiin laskettua huomattavasti uuden BLE-tekniikan (Bluetooth Low Energy) avulla, ja tietoturvaan tuli parannusta. Viimeisin Bluetooth versio 4.1 julkaistiin vuonna 2013, joka oli vain paranneltu versio edellisestä, 4.0-versiosta. [19.]

## 4.2 Arkkitehtuuri

Bluetoothin toiminta perustuu isäntä-orja-periaatteeseen. Periaate on sama kuin I2C-väylän toiminnassakin, eli yksi laite toimii isäntänä ja muut laitteet ovat orjia. Isäntä ei kuitenkaan millään tavalla poikkea muista Bluetooth-verkon laitteista, sillä jonkun on hoidettava isännän rooli verkossa. [18, s. 325.]

Pienin mahdollinen Bluetooth-verkko muodostuu jo pelkästään yhdellä isännällä ja orjalla. Tämä on havainnollistettu kuvan 5 kohdassa a. [18, s. 325.]



Kuva 5. Bluetooth-verkon topologiat

Bluetooth-verkosta, joka muodostuu yhdestä isännästä ja yhdestä orjasta, käytetään nimeä kaksipisteyhteys-verkko (Point-To-Point). Siinä kaksi laitetta lähettää toisilleen tietoa. Kuvassa 5 kohdassa b on käsitelty hieman laajempi Bluetooth-verkko, missä isännän kanssa samassa verkossa on useita orjia. Kohdassa c on useamman isännän muodostamia WPAN- (Wireless Personal Area Network) eli pikoverkkoja. Pikoverkkojen yhdistämää verkkoa kutsutaan nimellä scatternet. [18, s. 326.]

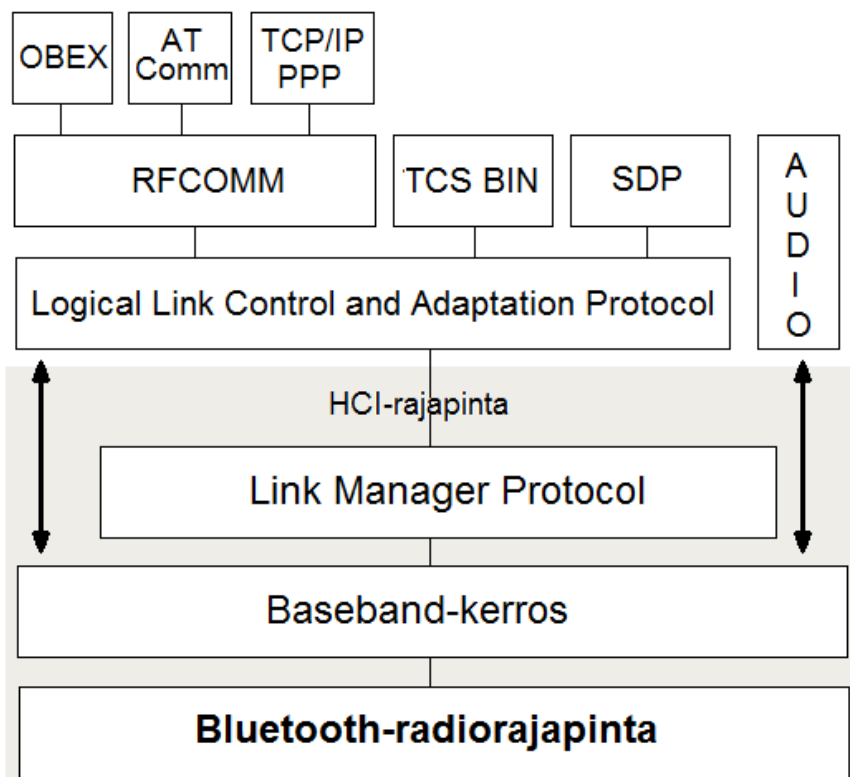
Scatternet-tyylisessä topologiassa yhdyskäytävän tehtävän verkkojen välillä hoitaa yksi verkon laitteista. Tämä laite voi myös hoitaa joko orjan tehtäviä kummassakin verkossa tai orjana toisessa ja isäntänä toisessa. Yhdyskäytävänä toimiva laite voi halutessaan keskustella vain

jommankumman osapuolen kanssa, joten kyse ei ole tavallisesta molempiin suuntiin toimivasta yhdyskäytävästä. [18, s. 326.]

Bluetooth-verkossa voi olla samassa solussa iso joukko passiivisia laitteita, vaikka aktiivisten laitteiden määrä solua kohden voi olla enimmillään kahdeksan. Isäntä voi halutessaan herättää jonkin verkon passiivista laitteista. [18, s. 326.]

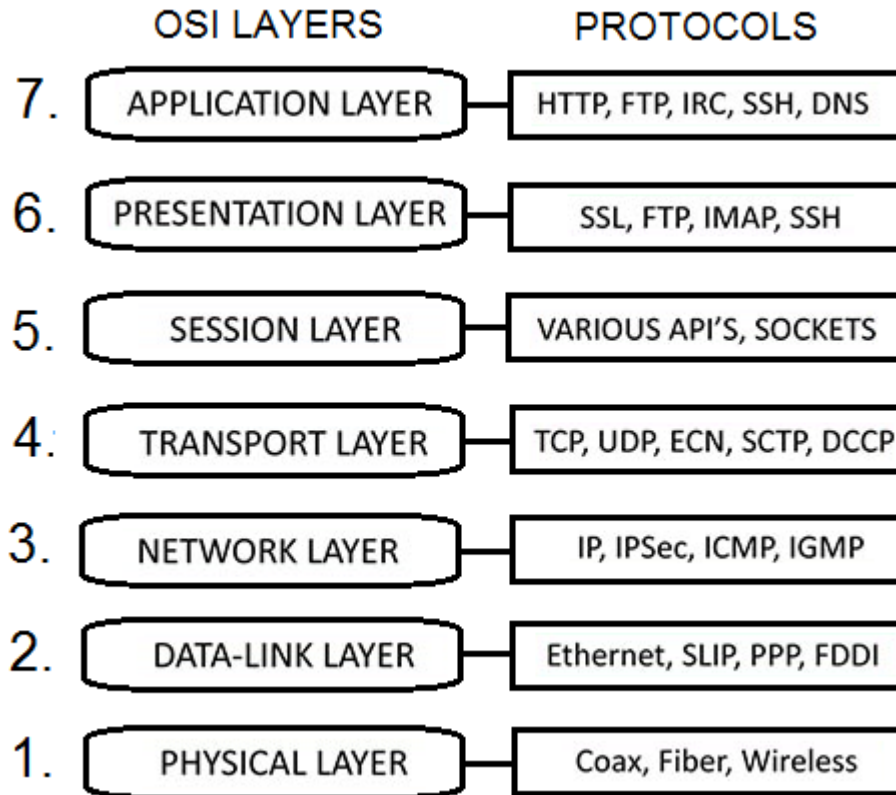
#### 4.2.1 Protokollapino

Bluetooth-määrittelyn mukainen protokollapino (kuva 6) käsittää sekä fyysisen kerroksen että siirtoyhteyskerroksen toiminnot. HCI-rajapinnan (Host Controller Interface) kautta kulkevan tietoliikenteen ohjauksen hoitaa isäntäkone, johon Bluetooth on sulautettu.



Kuva 6. Bluetooth-protokollapino [18, s. 327]

Bluetooth-protokollapino mukailee hieman OSI-protokollamallia (Open System Interconnection), joten sen voi osittain esittää OSI-protokollamallia (kuva 7) hyväksikäyttäen. [18, s. 327.]



Kuva 7. OSI-protokollamalli

Kuvassa 6 on esitetty Bluetooth-protokollapino. Radiorajapinta ja osa Baseband-kerrosta sijoittuvat OSI 1 -kerrokselle. LMP- (Link Manager Protocol) ja L2CAP-kerrokset (Logical Link Control and Adaptation Protocol) sisältävät OSI 2 -mallin toiminnot. Bluetooth ei siltikään noudata täysin OSI-protokollamallia, sillä OSI 1- ja OSI 2 -kerrosten raja kulkee keskeltä Baseband-kerrosta eikä Bluetooth-arkkitehtuuri noudata OSI-kerrosmallin periaatetta, jossa yksi kerros keskusteleekin vain viereisen kerroksen kanssa. Bluetooth-toiminnoissa voidaan ohittaa joitain OSI-mallin palveluja kuten puheen siirto. [18, s. 327.]

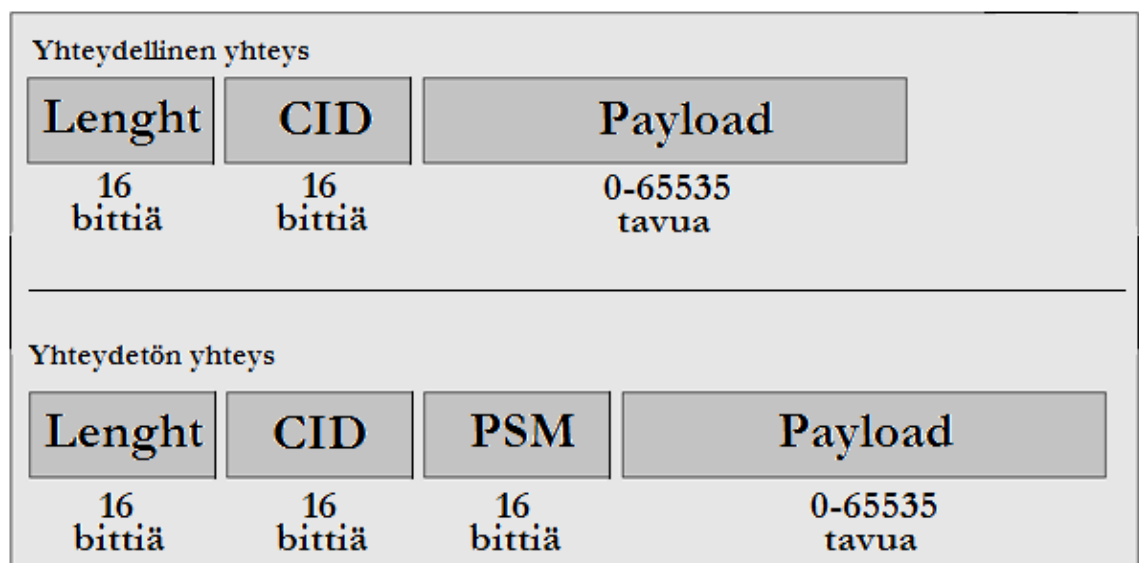
Yksinkertainen Bluetooth-ratkaisu ilman mitään palveluja, joka rajoittuu ylempiin kerroksiin HCI-rajapinnalla, saadaan yhteen moduuliin parin IC-piirin (Integrated Circuit) avulla käyttämällä Baseband-kerrosta, radiorajapintaa ja LMP:tä (Link Management Protocol). Moduuliin tarvitsee kytkeä enää vain virtalähde ja antenni, niin Bluetooth-ratkaisu on valmis. Ylemmät kerrokset ovat valmistajakohtaisia, ja ne vaativat Bluetooth SIG:n alaisuudessa toimivan BQRB:n (Bluetooth Qualification Review Board) BQP-sertifiointiprosessin (Bluetooth Qualification Program) läpäisemisen. [18, s. 327.]

#### 4.2.2 L2CAP-protokollakerros

L2CAP-protokollakerros on pakettimuotoinen ja asynkroninen tiedonsiirtoprotokolla, ja se sijoittuu HCI-rajapinnan yläpuolelle. L2CAP-protokollakerroksen (Logical Link Control and Adaptation) tiedonsiirrossa ei lähetettävälle datalle suoriteta ollenkaan virheentarkistusta. Mahdollisen tiedonsiirtovirheen sattuessa L2CAP-kerros ei osaa käsitellä virhettä, vaan virheidenkäsittelyn hoitaa Baseband-kerros. L2CAP-kerros vastaa OSI-protokollamallin siirtoyhteykskerrosta sekä osittain verkkokerrosta. [22, s. 310.]

L2CAP-kerroksen kaksi tärkeintä tehtävää ovat pakettien paloittelu ja kokoaminen, jotka sovellus on käsitellyt, sekä protokollapohjainen kanavointi. Baseband-kerros hoitaa paloittelujen sanomien ketjuttamisen siirtotiellä L\_CH-tunnisteilla, joita tässä ei käsitellä sen tarkemmin. [22, s. 310.]

Yhteydellisessä tiedonsiirrossa lähetettävään pakettiin lisätään otsikko, josta selviää paketin hyötykuorman (Payload) kokonaispituus sekä vastaanottajan kanavan tunniste (Channel ID). Yhteydetönmässä tiedonsiirrossa pakettiin lisätään vielä 16-bittinen PSM-kenttä (Protocol and Service Multiplexor). L2CAP-sanomien perusmuodot ovat nähtävissä kuvassa 8. [22, s. 311.] [22, s. 312.]



Kuva 8. L2CAP-sanomien perusmuodot [22, s. 312]

Datan siirron lisäksi L2CAP-kerrosten välillä on mahdollista siirtää merkinantosanomia. Merkinantosanoma koostuu kolmesta kentästä: sanoman pituus, kanavan tunniste ja hyötykuorma, joka sisältää erilaisia komentoja. [22, s. 312.]

### 4.3 Fyysinen kerros

Bluetooth-verkko toimii vapaalla ISM-alueella (Industrial Medical and Scientific), joka käyttää 2,4 GHz:n radiotaajuutta. Euroopan ISM-alue on taajuusvälillä 2400–2483,5 MHz, eli alueen laajuus on 83,5 MHz. Alue on jaoteltu 1 MHz:n välein 79 eri kanavaan, ja ensimmäinen kanava alkaa taajuudesta 2402 MHz. [18, s. 328.]

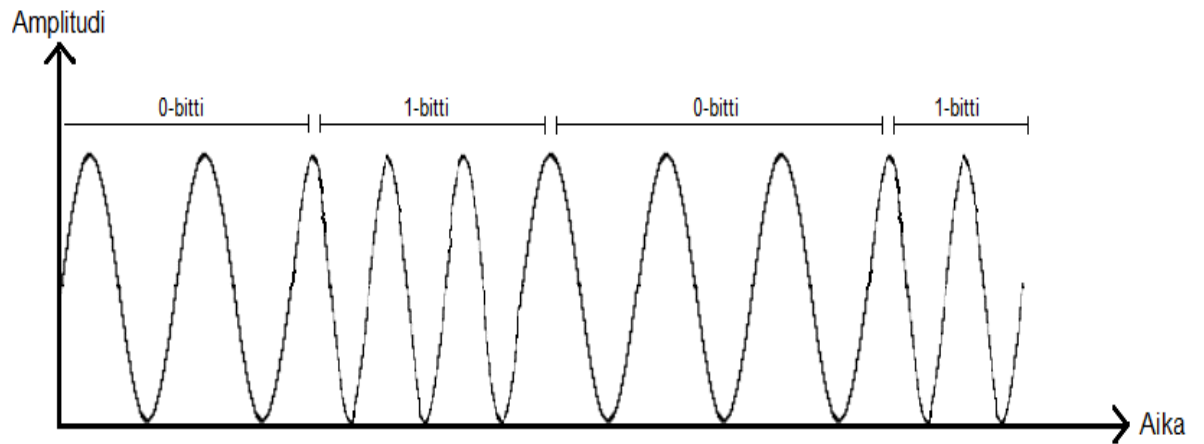
Bluetooth käyttää siis samaa taajuusaluetta ja kanavajakoa kuin WLAN-verkko. Bluetooth-verkon AFH-tekniikka pystyy välttämään yhteisten radiotaajuuksien käytön, jos WLAN-verkon solun alueella on käytössä vain yksi verkko. [18, s. 328.]

#### 4.3.1 Taajuusmodulointi

Taajuusmoduloinnilla tarkoitetaan kantoaallon modulointitapaa, jossa tiedonsiirto tapahtuu kantoaallon taajuuden vaihteluilla sen amplitudin pysyessä vakiona. [20].

Bluetooth käyttää tiedonsiirrossa juuri kyseistä taajuusmodulointi-menetelmää. 140–175 kHz kanavan keskitaajuutta ylempänä sijaitseva taajuus ilmaistaan 1-bittinä, ja 0-bitti saman 140–175 kHz kanavan keskitaajuuden alapuolella. Tämä on havainnollistettu kuvassa 9. [18, s. 328.]

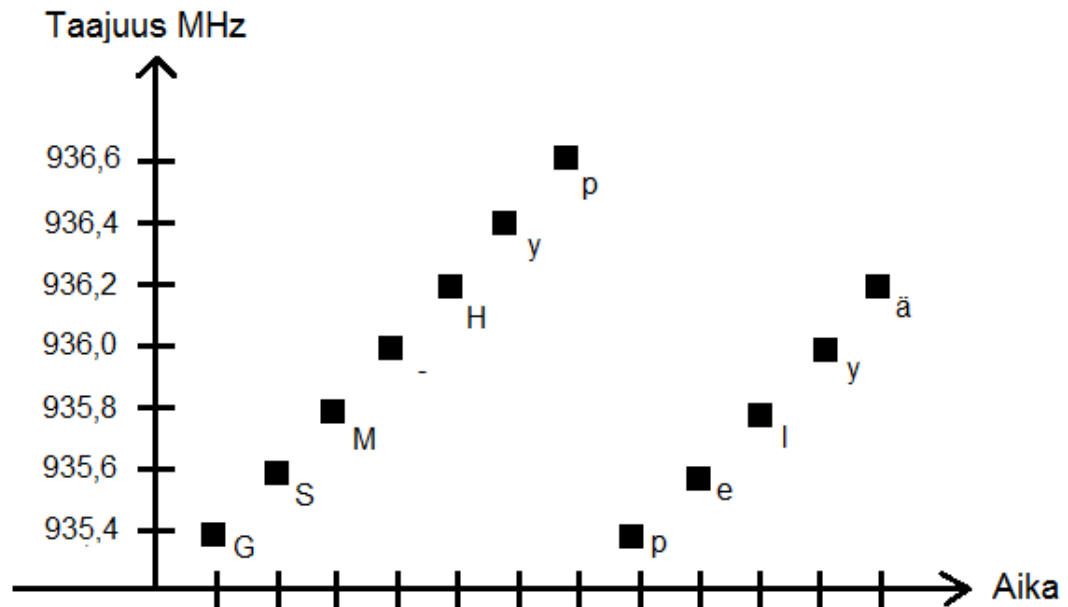




Kuva 9. Taajuusmodulointi

#### 4.3.2 Taajuushyppely

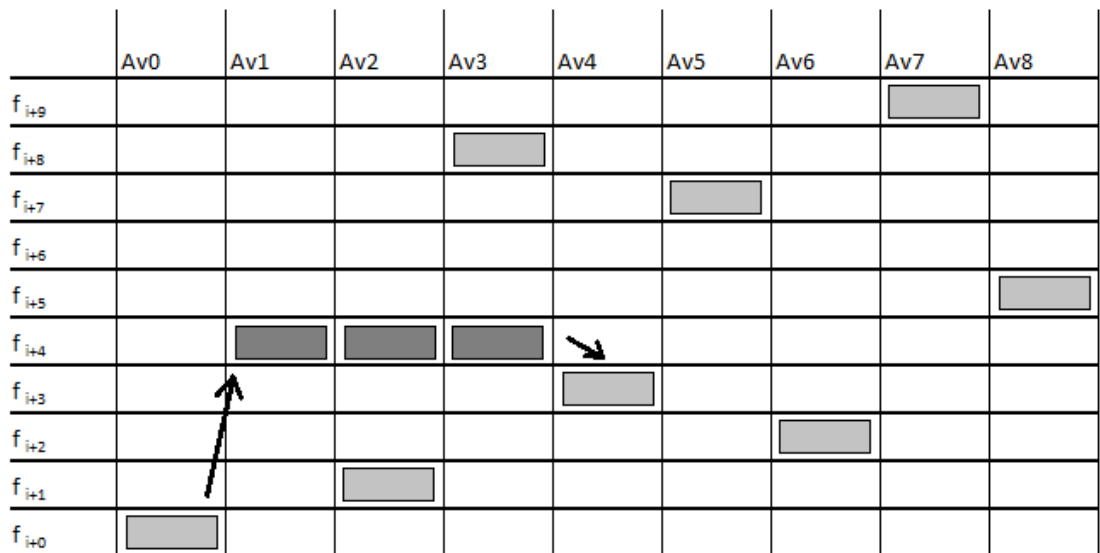
Taajuushyppely on lähinnä langattomissa järjestelmissä käytettävä hajaspektritekniikkaan perustuva menetelmä, joka alun perin kehiteltiin käytettäväksi sotilaselektronikassa. Taajuushyppelyn toimintaperiaate on seuraavanlainen: ”siirtoaika jaetaan kiinteän mittaisiin aikaväleihin, käytettävissä oleva siirtokaista jaetaan kiinteisiin alikanaviin ja sovitaan näennäissatunnaisesta hyppelyjärjestyksestä, jota sekä lähettäjä että vastaanottaja noudattavat”. Tiedonsiirron aikana tieto jaetaan purskeisiin, jonka jälkeen jokainen purske saa oman taajuuden ja hyppelyjärjestyksen, jonka perusteella ne lähetetään eteenpäin. Taajuushyppelyn periaate on nähtävissä kuvassa 10. [18, s. 117.]



Kuva 10. Taajuushyppelyn periaate [18, s. 117]

Jos hyppelytaajuuden aikavälin pituus on  $625 \mu\text{s}$  ja jokaisen aikavälin jälkeen taajuus muuttuu, saadaan hyppelytaajuudeksi 1600 hyppyä sekunnissa. Osa aikavälistä kuuluu radiolaitteiden taajuuden muuttamiseen oikeaksi, joten hyppelytaajuuden aikaväliä ei voida kokonaan käyttää tiedonsiirtoon. Tästä toimenpiteestä käytetään nimitystä suoja-aika, ja sen kesto on  $220 \mu\text{s}$ . Suoja-aika ei siis ole täysin merkityksetön, sillä se käyttää kolmasosan yhden taajuushyppelyn aikavälistä. [18, s. 328.]

Bluetooth-verkon taajuushyppelyssä on yksi erikoinen piirre verrattuna tavanomaisempaan taajuushyppelyyn, sillä siinä ne sanomat, joiden pituus ylittää yhden aikavälin, siirretään samalla taajuudella. Sitten hyppely siirtyy sille taajuudelle, minne se olisi alun perin jatkanut, jos edellinen siirrettävä sanoma olisi ollut korkeintaan yhden aikavälin pituinen. Taajuutta kuitenkin vaihdellaan ajoittain siten, että samalla taajuudella ollaan maksimissaan viiden sekunnin ajan. Bluetoothin taajuushyppely on nähtävissä kuvassa 11. [18, s. 117.] [18, s. 328.]



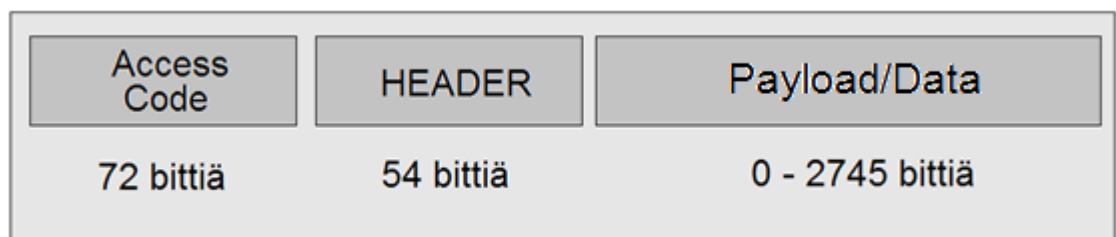
Kuva 11. Bluetooth-taajuushyppely [18, s. 329]

Kuvassa 11 on esitetty Bluetooth-taajuushyppely, jossa viestin lähettäminen jatkuu samalla taajuudella. Tämä koskee vain isäntää ja viestin vastaanottajaa/lähettäjä, joten muut laitteet jatkavat taajuushyppelyä annetun kaavan mukaan, sillä ne eivät ole tietoisia tästä. [18, s. 329.]

Isäntälaitteen osoite toimii hyppelyjärjestyksen muodostajana, joten kaikilla verkon laitteilla on oma hyppelyjärjestys, jos toistensa läheisyydessä on useampia pikoverkkoja. Samassa pikoverkossa olevat laitteet mukautuvat isäntänsä hyppelyjärjestykseen. [18, s. 329.]

#### 4.4 Viestikehykset

Bluetooth-verkon viestit noudattavat aina samaa perusmuotoa, jota on havainnollistettu kuvassa 12. [18, s. 333.]



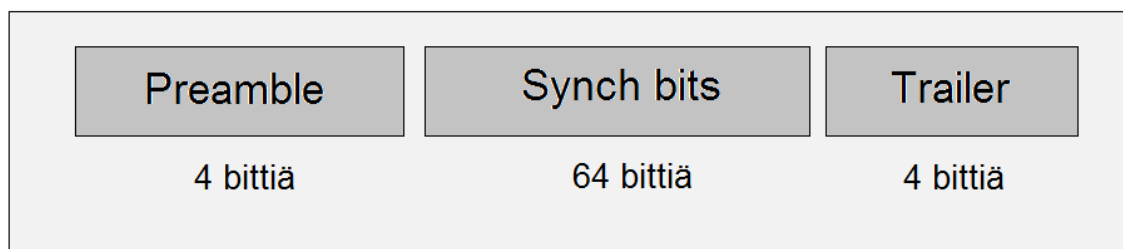
Kuva 12. Bluetooth-verkon viestikehyksen perusrakenne [18, s. 333]

Bluetoothin viestikehyksessä lähetetään aina vähintään kaksi kenttää: Access Code sekä Header eli otsikko. Payload eli data on viestikohtainen, ja se vaihtelee aina välillä 0–2745 bittä eli datakenttä voi olla tyhjä. [18, s. 334.]

#### Access Code

Access Code -kentän (kuva 13) ensimmäiset neljä bittiä ovat alkutahdistusbittejä (preamble). Tämän jälkeen tulee 64 bitin mittainen opetus- tai synkronointikenttä, ja lopussa on neljän bitin mittainen häntäbittikenttä (trailer). AC pitää myös huolen siitä, että saman pikoverkon asemat käyttävät samanlaista AC-kenttää. Synkronointikentän muodostamiseen käytetään isäntälaitteen 24-bittistä LAP-kenttää (Lower Address Part) niin, että synkronointisanoista muodostuville koodistoille saadaan sopivan mittainen Hamming-etäisyys. [18, s. 334.]

Alkutahdistus on 1010, jos synkronointikenttä alkaa 1-bitillä, ja synkronointikentän alkaessa 0-bitillä alkutahdistus on 0101. Tällä ehkäistään tasavirtakomponentin syntyminen vastaanotossa. Saman tasavirtakomponentin ehkäisemiseksi täytyy myös viimeiset neljä häntäbittiä koodata niin, että synkronointikentän päättyessä 0-bittiin neljä häntäbittiä saa arvot 1010. Jos synkronointikenttä päättyy 1-bittiin, häntäbitit saavat arvon 0101. [18, s. 334.]



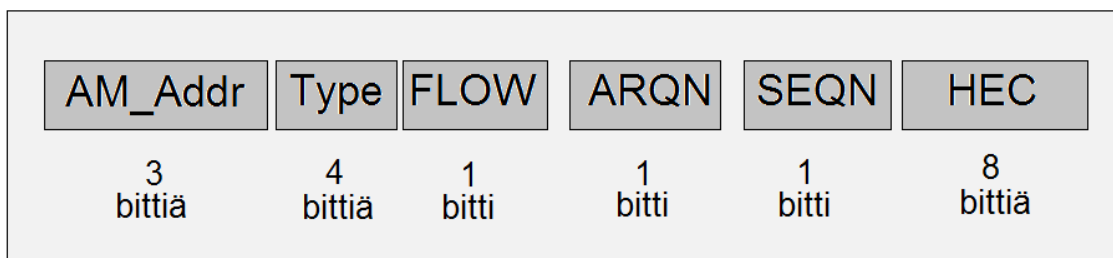
Kuva 13. Bluetooth-viestikehyksen Access Code -kenttä [21, s. 3]

#### Header – Otsikko

Otsikkokenttä (kuva 14) sisältää kaiken tiedon, joka sanoman vastaanottajan ja tyyppin tunnistamiseen tarvitaan. Bluetooth-verkon sallima laitteiden enimmäismäärä on kahdeksan, joten vastaanottajan tunnistamiseen käytetään kentän alussa kolmen bitin mittaista AM\_Addr-kenttää. Alavirtaan siirretty kolmen 0-bitin kenttä tarkoittaa yleislähetysanomaa, ja samat bitit ylävirtaan osoittavat isäntälaitteen. Otsikkokenttä sisältää myös vuon ohjauk-

seen, kuittaukseen sekä järjestysnumeroon liittyvää tietoa. Otsikkokentän suojauksessa on käytetty jakojäännöstarkistusta. [18, s. 334.]

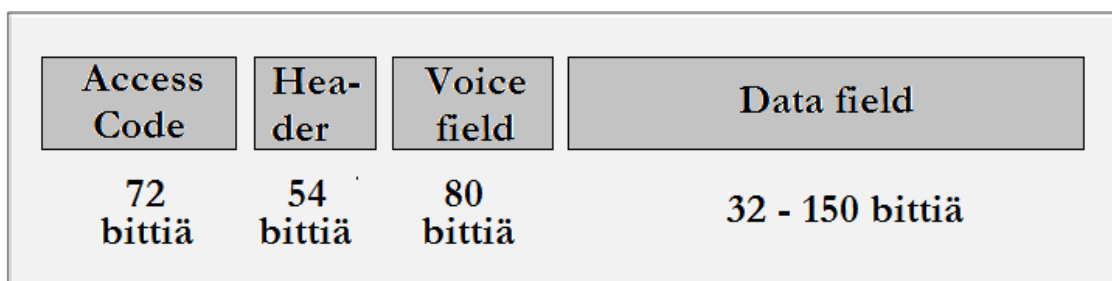
18-bittisen otsikkokentän siirtämiseen käytetään 1/3 FEC-suojausta (Forward Error Correction). Tällä tarkoitetaan siis sitä, että jokainen otsikkokentän bitti siirretään kolme kertaa eli otsikkokentän kokonaispituudeksi saadaan 54 bittiä. [18, s. 334]



Kuva 14. Bluetooth-viestikehyksen otsikkokenttä [22, s. 296]

Payload – Data

Payload- eli datakentässä (kuva 15) kulkee kolme perussanomaa: hallintasanomat, ACL-kanavan (Asynchronous Connection-Less) sanomat sekä SCO-kanavan (Synchronous Connection-Oriented) sanomat. Datakentän pituus on 2746 bittiä. Kuvassa 15 näkyy yhdistetty ääni- ja datasanoma, joka lähetetään kuten SCO-sanoma [18, s. 334.]



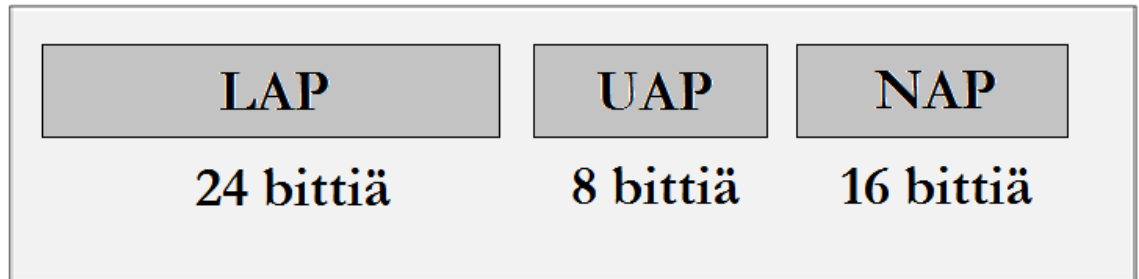
Kuva 15. Yhdistetty ääni- ja datasanomakenttä

#### 4.5 Laiteosoitteet

Jokaiseen Bluetooth-laitteeseen on kovakoodattu sen henkilökohtainen laiteosoite. Laiteosoitteessa käytetään MAC-osoitteen (Media Access Control) esitysmuotoa, ja se on 48 bittiä

pitkä, esitettyinä 12 luvun heksadesimaalisilla arvoilla. Laiteosoitteesta käytetään lyhennettä BD\_ADDR. [23.]

Bluetooth-laiteosoite koostuu kolmesta osasta: 24-bittinen LAP (Lower Address Part), 8-bittinen UAP (Upper Address Part) ja 16-bittinen NAP (Non-significant Address Part). Laiteosoite on nähtävissä kuvassa 16.



Kuva 16. Bluetooth-laiteosoite

LAP-osa on laitevalmistajan määrittelemä yksilöity laitteen tunnistusosa. UAP-osalla voidaan havaita korruptoituneita paketteja Bluetooth-tiedonsiirrossa. NAP-osalla ei ole mitään muuta merkittävää roolia kuin se on osana taajuushyppelyn synkronointikehystä. UAP- ja NAP-osat yhdessä muodostavat ns. yritys-ID:n, jolla tunnistetaan laitteen valmistaja. [23.]

## 5 KEHITYSYMPÄRISTÖN KÄYTTÖÖNOTTO

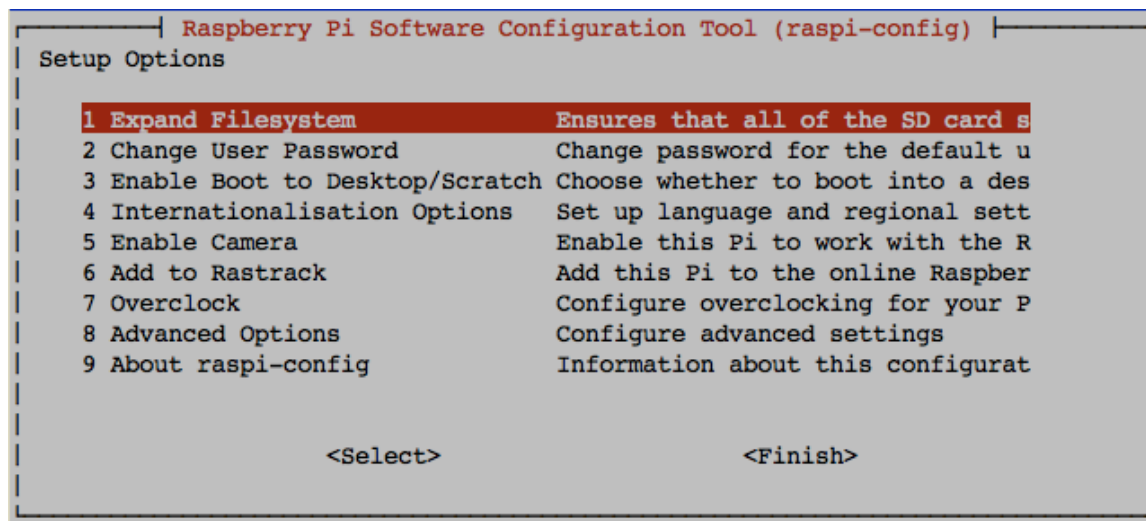
Kokonaisuudessaan kehitysympäristö koostuu kahdesta Raspberry Pi:stä ja niihin liitettävistä komponenteista sekä niiden kehittämiseen tarkoitettu kannettavasta tietokoneesta, jonka käyttöjärjestelmänä toimii 64-bittinen Ubuntu, versiolla 14.04 LTS. Kummankin Raspberry Pi:n käyttöjärjestelmänä on Raspbian Wheezy.

Varsinaisen ohjelmakoodin kehittäminen tapahtuu pääosin kannettavalla tietokoneella Eclipse-kehitysympäristön avulla ja C-ohjelmointikieltä käyttäen. Toteutettu ohjelma käännetään ristiinkääntäjällä SSH-yhteyden yli Raspberry Pi:lle. Käyttöliittymäsovelluksen kehittäminen tapahtuu suoraan Raspberry Pi:llä Qt Creatorin avulla C++-ohjelmointikieltä käyttäen.

### 5.1 SSH-yhteyden luominen

SSH eli Secure Shell, jota myös joskus kutsutaan nimellä Secure Socket Shell, on Unixiin pohjautuva komentorajapinta/protokolla, jolla on mahdollista saada turvallinen yhteys toiseen tietokoneeseen. Verkkoylläpitäjät käyttävät sitä paljon muun muassa servereiden ja verkon kontrollointiin. SSH koostuu kolmesta apuohjelmasta nimeltä slogin, ssh ja scp, jotka ovat suojattuja versioita aikaisemmista Unixin rlogin-, rsh- ja rcp-apuohjelmista. SSH-komennot ovat salattuja sekä suojattuja usealla eri tavalla. Asiakas- ja palvelinyhteydet todennetaan digitaalisella sertifiikatilla kummastakin päästä, ja salasanat ovat hyvin salattuja. [24.]

Ensimmäinen vaihe SSH-yhteyden luomiseen on sallia SSH-yhteys Raspberry Pi:stä. Tämä onnistuu Raspberry Pi:n asetuksista, joihin pääsee käsiksi kirjoittamalla komentoriville komennon ”sudo raspi-config”. Tämän jälkeen avautuu kuvan 17 mukainen valikko, josta valitaan **8 Advanced Options**, ja edelleen **A4 SSH**, josta valitaan **Enable**.



Kuva 17. Raspberry Pi:n asetukset-ikkuna

Seuraava vaihe on kytkeä Ethernet-kaapeli kannettavan tietokoneen ja Raspberry Pi:n välille. Ubuntuun työpöydän oikeasta yläkulmasta painetaan verkkoyhteyskuvan päältä oikealla hiirenpainikkeella, ja valitaan **Muokkaa yhteyksiä**. Ikkunan avauduttua valitaan **Ethernet**-otsikon alta **Kiinteä yhteys** ja **Muokkaa**. Seuraavan ikkunan avauduttua valitaan **IPv4-asetukset**-välilehden alta **Menetelmä**-valikosta valinta **Jaettu muille tietokoneille**. Nyt tietokone onnistuu löytämään Raspberry Pi:n.

Kirjoittamalla Ubuntuun komentoriville komento ”ifconfig” saadaan tietoon Raspberry Pi:n IP-osoite (Internet Protocol). Eth0 inet addr -kohdasta voidaan nähdä, että Ubuntu näkee Raspberry Pi:n IP-osoitteen olevan 10.42.0.1. Tämän jälkeen täytyy Raspberry Pi:n IP-osoite muokata oikeaksi. Tämä onnistuu esimerkiksi käyttämällä Raspberry Pi:n SD-muistikorttia kortinlukijassa, jolloin päästään käsiksi Raspberry Pi:n cmd-tiedostoon. Cmdline.txt-tekstitiedoston loppuun lisätään ”ip=10.42.0.2”. IP-osoitteen viimeisen luvun saa muuttaa, kunhan ei ylitä aliverkon peitettä. Kuvassa 18 on alleviivattu ifconfig-komennolla saatu osoite, jota tarvitaan SSH-yhteyden luomiseen.

SSH-yhteys Ubuntuun ja Raspberry Pi:n välille saadaan kirjoittamalla Ubuntuun komentoriville komento ”sudo ssh pi@10.42.0.2”. Jos etäyhteys toimii, Raspberry Pi kysyy tunnuksia, ja pääsee kirjautumaan sisään. SSH-yhteyden saa Raspberry Pi:lle myös Eclipse-kehitysympäristön kautta, jota onkin tässä työssä käytetty Ubuntuun komentorivin sijaan.



```

jakeboi@ubuntumylly: ~
jakeboi@ubuntumylly:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 00:1d:72:e0:69:30
          inet addr:10.42.0.1  Bcast:10.42.0.255  Mask:255.255.255.0
          inet6 addr: fe80::21d:72ff:fee0:6930/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:43 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:7716 (7.7 KB)
          Interrupt:17

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:209 errors:0 dropped:0 overruns:0 frame:0
          TX packets:209 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:16696 (16.6 KB)  TX bytes:16696 (16.6 KB)

wlan0    Link encap:Ethernet  HWaddr 00:17:c4:51:fd:b9
          inet addr:192.168.0.13  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::217:c4ff:fe51:fdb9/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1

```

Kuva 18. Ubuntu ifconfig-komento

## 5.2 Ristiinkääntäjän asennus

Tietokoneen perus-GCC-kääntäjä (The GNU Compiler Collection) on tarkoitettu toimimaan ja kääntämään ohjelmakoodia Intel-arkkitehtuurille. Toisin sanoen se luo käännetyistä ohjelmakoodista binääritiedostoja, joiden on tarkoitus toimia samalla prosessoriarkkitehtuurilla. Kuten Raspberry Pi:n Linux-pohjaisissa käyttöjärjestelmissä, GCC-kääntäjällä käännetään ohjelmakoodi toimimaan ARM-arkkitehtuurille. [25.]

Ristiinkääntäjällä tarkoitetaan kääntäjää, joka kääntää ohjelmakoodia eri prosessoriarkkitehtuurilta toiselle. Esimerkiksi ristiinkääntäjä voi toimia Intel-arkkitehtuurin tietokoneessa, mutta luokin binääritiedostoja, jotka toimivat ARM-arkkitehtuurilla. Tällä tavalla on mahdollista kehittää ohjelmakoodia omalla työpöytäkoneella ja ajaa binääritiedosto esimerkiksi juuri Raspberry Pi:lle. [25.]

Miksei siis käytettäisi Raspberry Pi:tä ohjelmakoodin kehittämiseen ja käännettäisi koodia Raspberry Pi:n omalla GCC-kääntäjällä? Raspberry Pi on varmasti nopeampi kuin tavalliset mikrokontrollerit, mutta siinä on kuitenkin melko rajallinen määrä keskusmuistia, ja nykyajan

perustyöpöytäkoneet ovat selvästi nopeampia kuin Raspberry Pi. Työpöytäkoneella on mahdollista käyttää ohjelmakoodin kätevään kehittämiseen tarkoitettuja kehitysympäristöjä, jotka ovat yleensä liian raskaita Raspberry Pi:n tasoiselle laitteelle. Tässäkin työssä käytettävän Eclipse-kehitysympäristön käyttö on huomattavasti sulavampaa omalta työpöytäkoneelta kuin Raspberry Pi:ltä. [25.]

Ristiinkääntäjän asentaminen Ubuntuille lähtee liikkeelle asentamalla GIT-työkalu, jolla on mahdollista ladata klooni ristiinkääntäjätyökalupaketista GitHub.com-versionhallintasivustolta. GIT-työkalun saa kirjoittamalla komentoriville komennon ”sudo apt-get install build-essential git”. [25.]

Ristiinkääntäjä tarvitsee 32-bittiset kirjastotiedostot toimiakseen oikein 64-bittisessä Ubuntuissa. 12.04- ja vanhemmalla Ubuntu-versiolla kirjastotiedostot saadaan komennolla ”sudo apt-get install ia32-libs”. 13.10- ja uudemmalla Ubuntu-versiolla kirjastotiedostot saadaan komennolla ”sudo apt-get install libc6:i386 libgcc1:i386 gcc-4.6-base:i386 libstdc++5:i386 libstdc++6:i386 lib32z1 lib32ncurses5 lib32bz2-1.0 lib32asound2”. [25.]

Seuraavaksi luodaan kansio kotihakemistoon, jolle voi antaa nimeksi vaikkapa rpi, komennolla ”sudo mkdir rpi”. Siirrytään rpi-kansioon, ja kirjoitetaan komento, jolla ladataan GitHubista klooni Raspbianin virallisesta ristiinkääntäjän työkalupaketista, ”sudo git clone git://github.com/raspberrypi/tools.git”. [25.]

### 5.3 Bcm2835-kirjaston asentaminen Raspberry Pi:lle

Raspberry Pi käyttää Broadcomin piiriä GPIO-pinnien ohjaukseen. Jotta nämä pinnit saataisiin käyttöön C-ohjelmointikielellä, täytyy Raspberry Pi:lle asentaa bcm2835-ohjelmakirjasto.

Kirjaston asentaminen alkaa menemällä /home/pi-kansioon kirjoittamalla komentoriville komento "cd /home/pi". Seuraavaksi ladataan bcm2835-kirjasto komennolla "sudo wget http://www.airspayce.com/mikem/bcm2835/bcm2835-1.36.tar.gz". Tiedosto puretaan komennolla "sudo tar zxvf bcm2835-1.36.tar.gz". Puretut tiedostot löytyvät kansioista nimeltä /home/pi/bcm2835-1.36. Kansioon päästään komennolla "cd bcm2835-1.36". Seuraavaksi täytyy ajaa configure.exe komennolla "sudo ./configure", jonka jälkeen suoritetaan makefile komennolla "sudo make". Lopuksi ajetaan komennot "sudo make check" ja "sudo make install", jonka jälkeen kirjasto on käyttövalmis.

Kirjaston käyttöönottoaminen onnistuu sisällyttämällä bcm2835-header -tiedosto projektiin eli kirjoittamalla projektin alkuun koodirivi #include bcm2835.h.

### 5.4 Eclipse-kehitysympäristö

Eclipse on alustariippumaton ja avoimeen lähdekoodiin perustuva kehitysympäristö, jota käytetään ohjelmakoodin kehittämiseen. Eclipse on luotu käyttäen Java-ohjelmointikieltä, ja Eclipseä käytetäänkin useimmiten juuri Java-ohjelmointikielen kehittämiseen. Eclipsellä onnistuu kuitenkin myös useimpien yleisimpien ohjelmointikielien kehittäminen, kuten C- ja C++-ohjelmointikielien.

Eclipse-kehitysympäristön asetuksiin täytyy tehdä tiettyjä muokkauksia, jotta ristiinkääntäjä, SSH-yhteys Raspberry Pi:lle kehitysympäristön kautta ja bcm2835-kirjasto kehitysympäristössä saataisiin toimimaan.

### 5.4.1 Eclipsen asennus

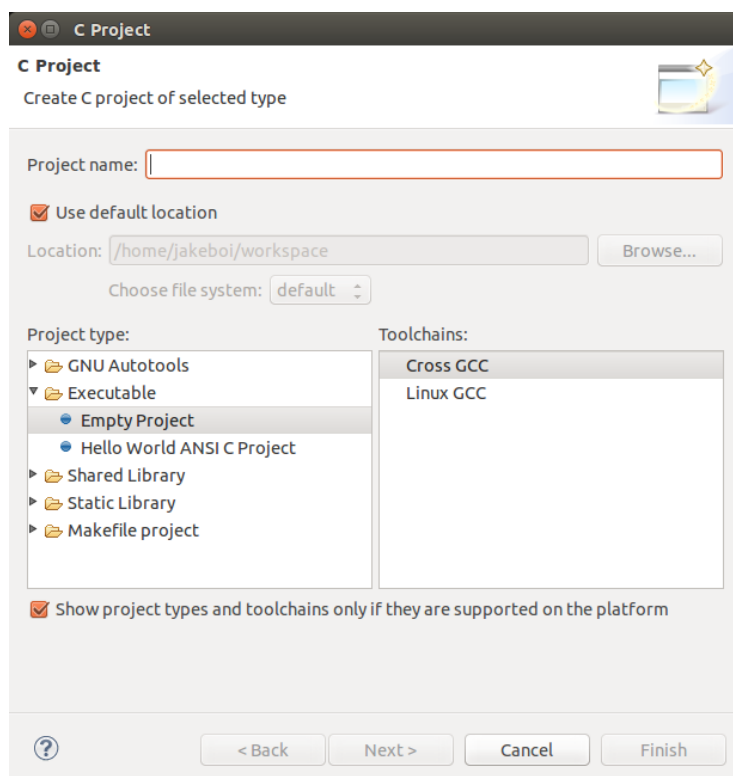
Ennen kuin asennetaan Eclipse, täytyy asentaa Java. Java asennetaan kirjoittamalla Ubuntun komentoriville komento ”sudo apt-get install openjdk-7-jre”. Viimeisin Eclipse-paketti voidaan ladata osoitteesta <http://www.eclipse.org/downloads/?osType=linux>.

Ladattu eclipsen .tar-tiedosto voidaan siirtää haluttuun hakemistoon, jonka jälkeen se puretaan komennolla ”sudo tar -xvf *filename.tar.gz*”.

Eclipse avataan komentorivin kautta menemällä hakemistoon, jossa Eclipsen .exe-tiedosto on, ja käynnistetään se sudo-oikeuksilla eli komennolla ”sudo ./eclipse”.

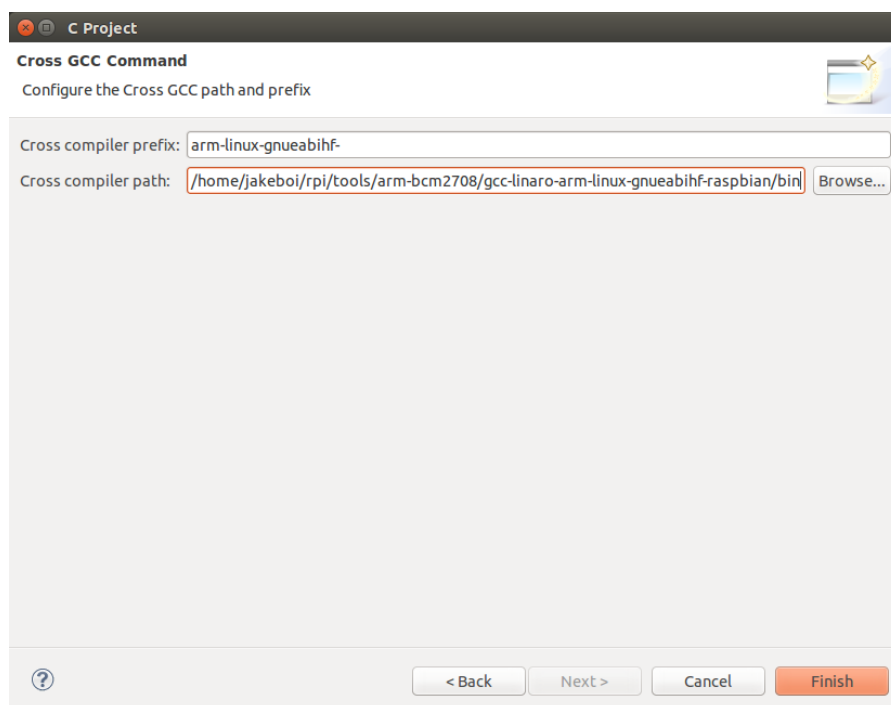
### 5.4.2 Ristiinkääntäjän käyttöönotto

Ristiinkääntäjä otetaan käyttöön Eclipsessä uuden projektinluonnin yhteydessä. **File -> New -> C Project. Project type-**valikosta **Executable -> Empty Project** ja **Toolchains-**valikosta **Cross GCC**. Eclipsen projekti-ikkuna on nähtävissä kuvassa 19.



Kuva 19. Eclipsen C-projekti-ikkuna

Seuraavaksi valitaan ikkunan alareunasta kahdesti **Next**, niin kuvan 20 mukainen ikkuna tulee esiin. **Cross compiler prefix** -kohtaan kirjoitetaan ristiinkääntäjän nimi eli arm-linux-gnueabihf-. **Cross compiler path** -kohtaan **Browse**-painikkeesta valitaan ristiinkääntäjän polku: /home/yourusername/rpi/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian/bin. Lopuksi painetaan **Finish**.



Kuva 20. Ristiinkääntäjän konfigurointi-ikkuna

#### 5.4.3 Bcm2835-kirjaston käyttöönotto Eclipse-projektissa

Kun Raspberry Pi:lle on asennettu bcm2835-kirjasto, kirjaston .a-tiedosto eli varsinainen kirjastotiedosto löytyy oletuksena kansiota /usr/local/lib ja .h-tiedosto eli header-tiedosto löytyy kansiota /usr/local/include. Libbcm2835.a-tiedosto täytyy käydä lisäämässä Eclipse-kehitysympäristössä kääntäjän linkkerille, ja bcm2835.h-tiedosto kääntäjän include-kansioon, jotta kirjaston funktiot ja toiminnot saadaan käyttöön luoduissa projekteissa.

Kumpikin tiedosto kopioidaan esimerkiksi muistitikulle tai siirretään ne SSH-yhteyden kautta. Libbcm2835.a tiedosto täytyy käydä viemässä Eclipsen kääntäjän lib-kansioon: home/rpi/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian/arm-linux-gnueabihf/lib. Bcm2835.h-tiedosto täytyy käydä viemässä kääntäjän include-kansioon:

/home/yourusername/rpi/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian/arm-linux-gnueabihf/libc/usr/include.

Eclipsessä libbcm2835.a-kirjaston lisääminen linkkerille projektiin tapahtuu seuraavasti: Klikataan hiiren oikealla projektin päältä ja valitaan **Properties**.

-> **C/C++ Build -> Settings**

-> **Gross GCC Linker -> Libraries**

Valitaan **Libraries (-l)** -kohdasta **Add**, ja kirjoitetaan tekstikenttään **bcm2835** ja painetaan **OK**. Lopuksi valitaan **Apply** ja **OK**.

Include-kansion, joka sisältää myös kaikki muut tärkeimmät C-ohjelmoinnissa tarvittavat header-tiedostot, projektiin lisääminen tapahtuu seuraavasti: Klikataan hiiren oikealla projektin päältä ja valitaan **Properties**.

-> **C/C++ General -> Paths and Symbols**

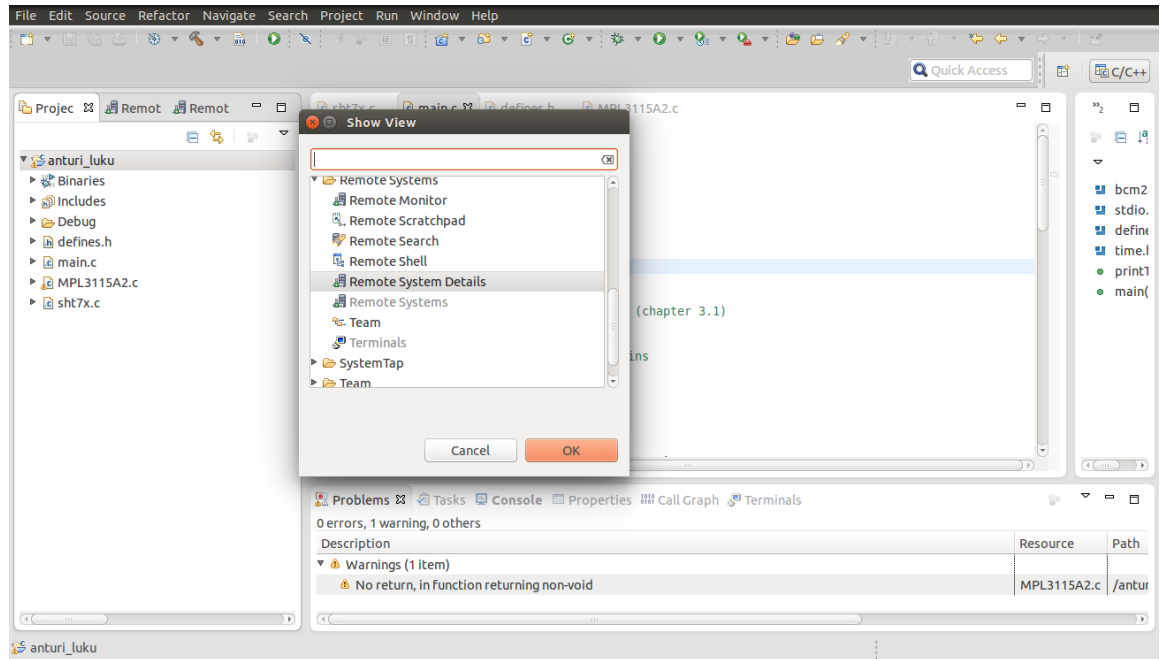
-> **Languages -> GNU C**

Valitaan **Include directories** -kohdasta **Add** ja **File System** ja haetaan polku: /home/yourusername/rpi/arm-bcm2708/gcc-linaro-arm-linux-gnueabihf-raspbian/arm-linux-gnueabihf/libc/usr/include. Paina **OK**, **OK**, **Apply** ja **OK**.

#### 5.4.4 SSH-yhteys Raspberry Pi:lle

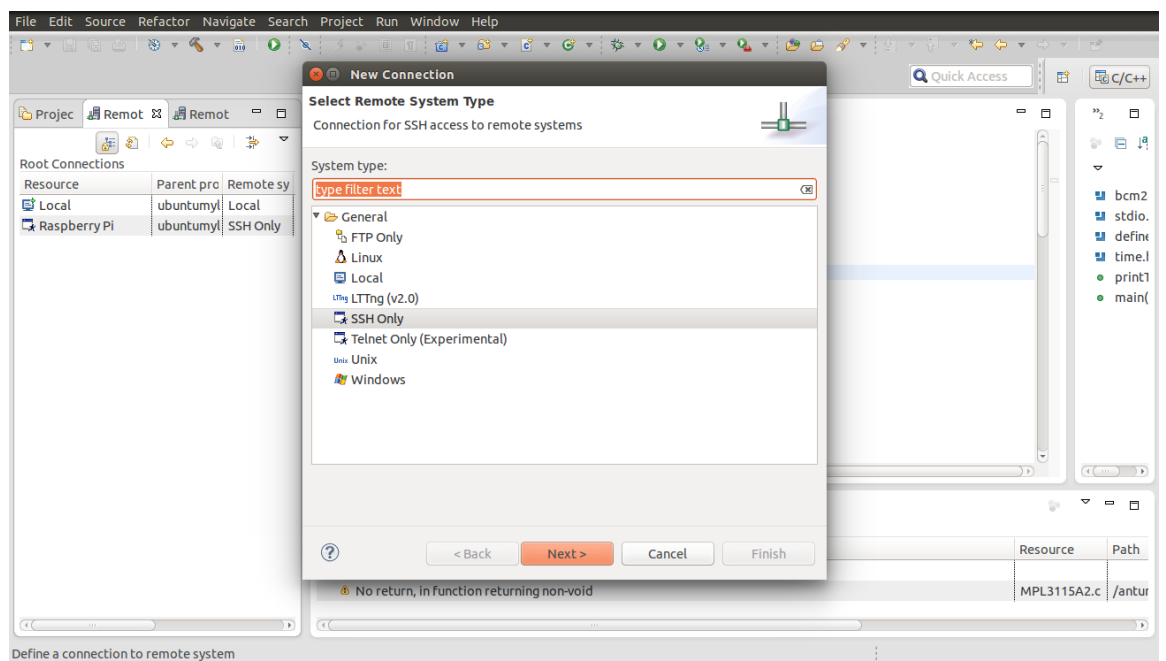
Tässä opinnäytetyössä SSH-yhteyttä ei oteta Raspberry Pi:lle Ubuntun komentorivin kautta, vaan Eclipse-kehitysympäristöstä, jota kautta käännetyn ohjelmakoodin binääritiedosto on erittäin kätevä siirtää Raspberry Pi:lle.

Ensimmäinen vaihe on laittaa virrat Raspberry Pi:lle ja yhdistää Ethernet-kaapeli tietokoneen ja Raspberry Pi:n välille. Klikataan hiiren oikealla painikkeella **Window -> Show View -> Other**. Silloin aukeaa kuvan 21 mukainen ikkuna. Valitaan ikkunasta **Remote Systems**-kansio ja **Remote System Details** ja painetaan **Ok**.



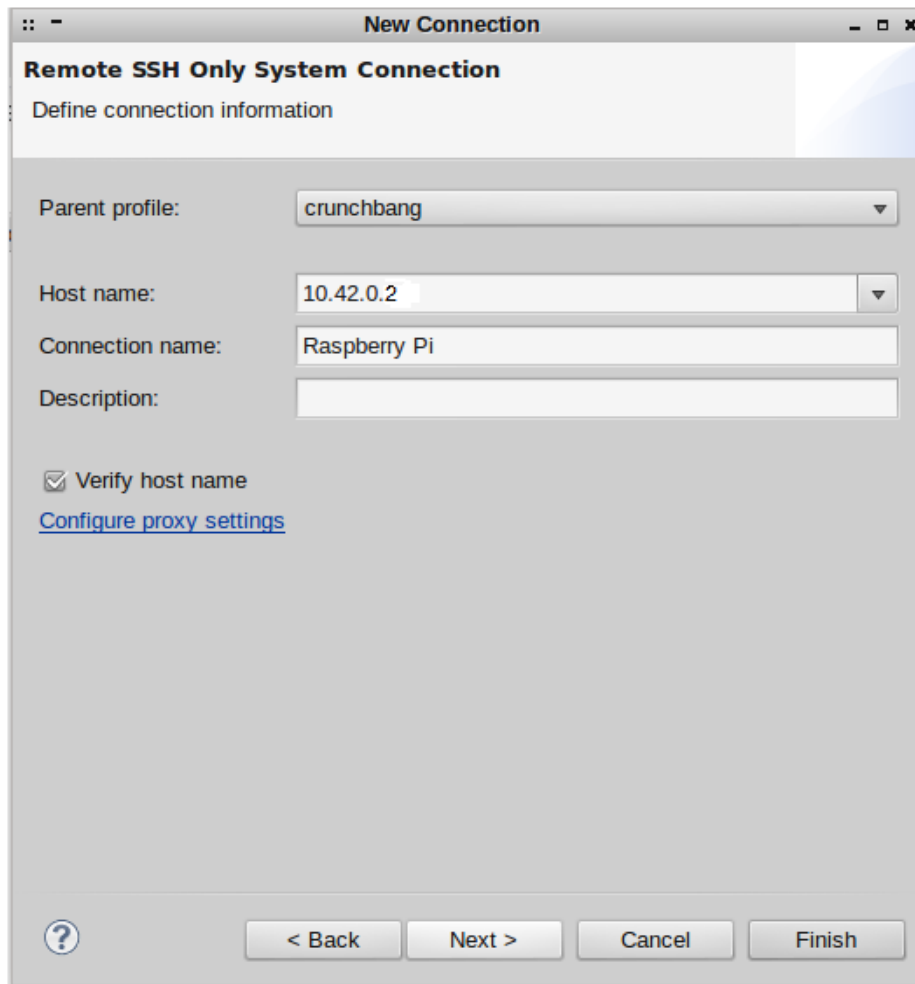
Kuva 21. Remote System -näkymä

Jos **Remote Systems Details** avautuu työtilan alareunaan, voidaan se vetää samaan paikkaan työtilan vasempaan reunaan, jossa **Project Explorer** -välilehtikin sijaitsee. Hiiren oikealla klikataan **Remote Systems Details** -aluetta, ja valitaan **New Connection**. Avautuu kuvan 22 mukainen ikkuna, josta valitaan Remote System -tyypiksi **SSH Only** ja **Next**.



Kuva 22. Remote System -tyypin valintaikkuna

**Remote SSH Only System Connection** -ikkunan (kuva 23) avauduttua kirjoitetaan Raspberry Pi:n IP-osoite **Host name** -kohtaan. **Connection name** -kohtaan voi laittaa haluamansa yhteysnimen, joka tässä tapauksessa on Raspberry Pi. Lopuksi painetaan **Finish**.

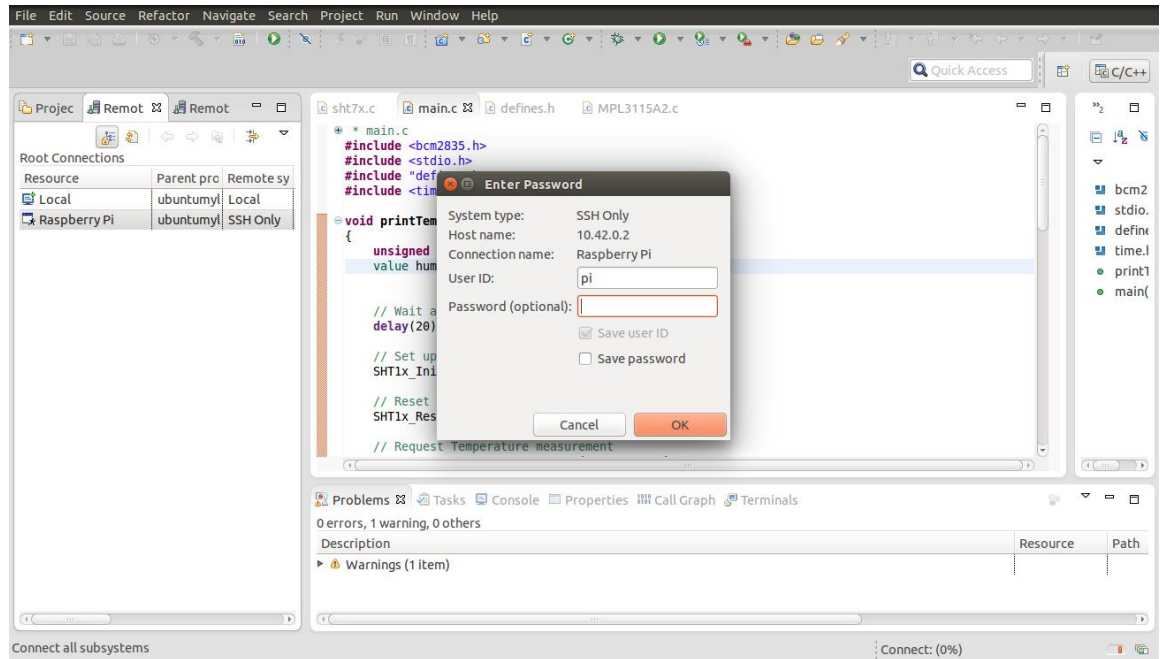


Kuva 23. Remote SSH Only System Connection -ikkuna

Tässä vaiheessa ilmaantuu toinen laite, nimeltä Raspberry Pi, **Remote System Detail** -välilehdelle, työtilan vasempaan reunaan. Klikataan hiiren oikealla **Resourcen** alta **Raspberry Pi** -kohdasta ja valitaan **Connect**.

Seuraavaksi avautuu ikkuna (kuva 24), johon laitetaan käyttäjänimi kohtaan **User ID** ja salasana kohtaan **Password**. Lopuksi painetaan **OK**, jonka jälkeen saattaa tulla varoitusikkuna, jossa kysytään varmistusta yhteyden luomiseen. Hyväksytään yhteyden luominen, jonka jälkeen SSH-yhteyden pitäisi toimia Eclipsen ja Raspberry Pi:n välillä.





Kuva 24. SSH-yhteys Raspberry Pi:lle

Klikataan hiiren oikealla uudestaan **Remote Systems Detail** -välilehdestä **Resourcen** alta **Raspberry Pi** -kohdasta, ja valitaan **Show in Remote Systems View**. Tällä avautuu samaan paikkaan työtilan vasempaan reunaan välilehti nimeltä **Remote Systems**, jossa on mahdollista navigoida sekä työpöytäkoneen tiedostojärjestelmässä että Raspberry Pi:n tiedostojärjestelmässä. Tällä tavalla Eclipse:ssä käännetyn ohjelmakoodin binääritiedosto on kätevä siirtää Raspberry Pi:lle ajettavaksi.

Terminaalin saa auki klikkaamalla hiiren oikealla painikkeella, **Remote Systems** -välilehdellä **Raspberry Pi**:n alta kohdasta **Ssh Terminals, Launch Terminal**. Tämä avaa terminaali-ikkunan Eclipse-työtilan alareunaan.

## 5.5 Qt Creator-kehitysympäristö

Qt Creator on alustariippumaton graafisten käyttöliittymien suunnitteluun tarkoitettu kehitysympäristö, jota on mahdollista ohjelmoida C++-, JavaScript- tai QML-ohjelmointikielellä (Qt Meta Language). Qt Creator käyttää Linux-käyttöjärjestelmissä GNU-kääntäjäpaketin C++-kääntäjää ohjelmakoodin kääntämiseen.

### Qt Creatorin asennus ja asetukset

Ensimmäisenä ladataan Qt Creatorin vaatimat Qt-kehitystyökalut kirjoittamalla Raspberry Pi:n komentoriville komento ”sudo apt-get install qt4-dev-tools”. Itse Qt Creator ladataan komennolla ”sudo apt-get install qtcreator”.

Qt Creatorin asetuksista otetaan GCC ”toolchain” käyttöön valitsemalla ikkunan yläreunasta valikko **Tools -> Options -> Build & Run**.

**Tool Chains** -välilehdeltä valitaan **Add** ja **GCC**.

Valitaan kääntäjälle polku: /usr/bin/arm-linux-gnueabi-gcc-4.6

Debuggerille polku: /usr/bin/gdb

Mkspec: Default

Jotta projekti saadaan näkyään työpöytäprojektina, valitaan ikkunan yläreunasta **Help -> About Plugins -> Device support**.

Otetaan rasti pois kohdasta **Remote Linux** ja käynnistetään Qt Creator uudestaan.

Valitaan ikkunan yläreunasta **Tools -> Options -> Build & Run**.

**Qt version** -kohtaan haetaan polku /usr/bin/qmake-qt4.

## 6 TYÖN SUORITUS

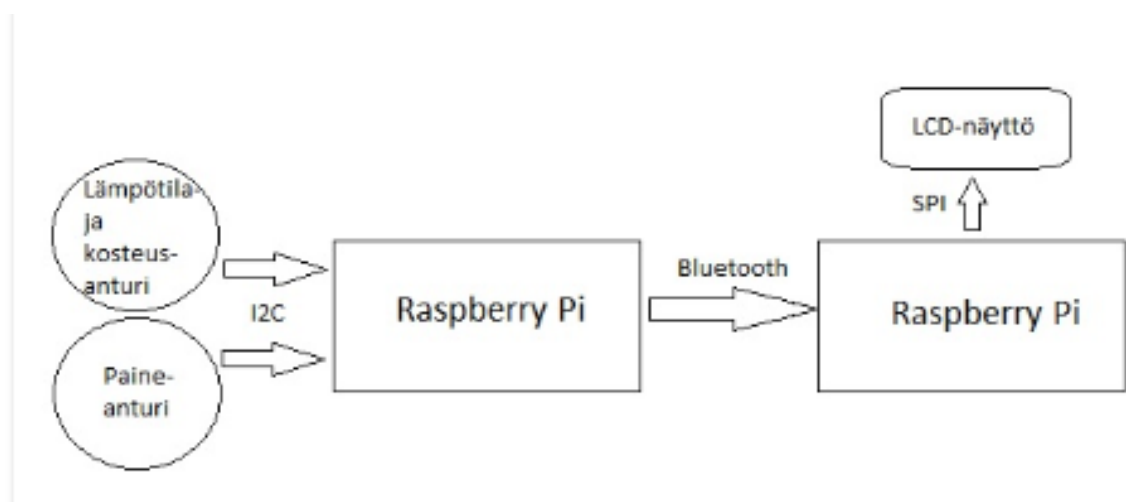
Tämän opinnäytetyön käytännön työ koostuu kahdesta Raspberry Pi:stä, kahdesta eri anturista (ja yhdestä ulkoisesta vastuksesta), LCD-näytöstä sekä järjestelmälle toteutettavasta ohjelmakoodista. Järjestelmän ohjelmakoodi kehitetään pääosin Eclipsen avulla. Qt Creatorilla kehitetään järjestelmän käyttöliittymän ohjelmakoodi.

Ensimmäiseen Raspberry Pi:hin on kytketty kaksi anturia, joista toisella mitataan lämpötilaa sekä ilmankosteutta, ja toisella mitataan ilmanpainetta. Data liikkuu antureilta Raspberry Pi:lle I2C-väylän kautta.

Antureita lukevalta Raspberry Pi:ltä tuleva data siirretään langattomasti Bluetoothin avulla toiselle Raspberry Pi:lle. Toisella Raspberry Pi:llä tehdään systeemin käyttöliittymä, jonka avulla mittausdata saadaan näkyviin Raspberryn GPIO-porttiin kytkettyyn, SPI-väylää hyväksikäyttävään, pieneen LCD-näyttöön.

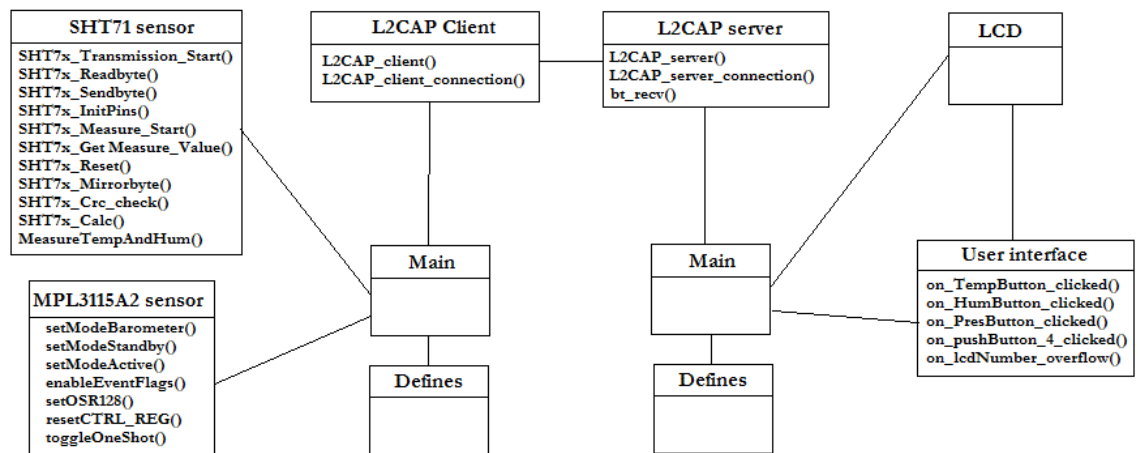
Antureiksi valittiin tarkoituksella digitaaliset, jotta välttyttäisiin A/D-muuntimen (Analog to Digital) käytöltä, ja saataisiin työ pidettyä täysin digitaalisena.

Koko systeemin lohkokaavio on esitetty kuvassa 25.



Kuva 25. Systeemin lohkokaavio

Kuvassa 26 on nähtävissä järjestelmälle toteutetun ohjelmakoodin rakennetta kuvaava kaavio. Kaaviossa on nähtävissä eri osioissa toteutetut funktiot. SHT71 sensor -osiossa on kuvattu lämpötila- ja ilmankosteusanturin funktiot, josta data välittyy main-funktioon. MPL3115A2 sensor -osio kuvaa paineenmittauksessa käytettyjä funktioita, josta data välitetään myös main-funktiolle. Main-funktiosta data välitetään Bluetooth-asiakkaalle, josta taas eteenpäin Bluetooth-palvelimelle. Data on tullut nyt toiseen main-osioon, josta data välitetään suoraan LCD-osioon. Jos lopullisen ohjelmakoodin toiminta olisi ollut tavoitteiden mukainen, olisi main-osioista siirrytty ensin käyttöliittymä-osion kautta LCD-osioon. Defines-osiot sisältävät muun muassa esikäännettävien otsikkotiedostot, vakiot ja funktioiden esittelyt.



Kuva 26. Järjestelmälle toteutetun ohjelmakoodin rakennetta kuvaava kaavio

## 6.1 Lämpötilan ja kosteuden mittaus

Lämpötilan ja ilmankosteuden mittauksessa päätettiin käyttää suosittua Sensirion SHT7x -sarjan SHT71-anturia, johon on integroitu sekä lämpötilan että ilmankosteuden mittaukseen tarvittava piiri, joten selvittää ainoastaan yhdellä anturilla.

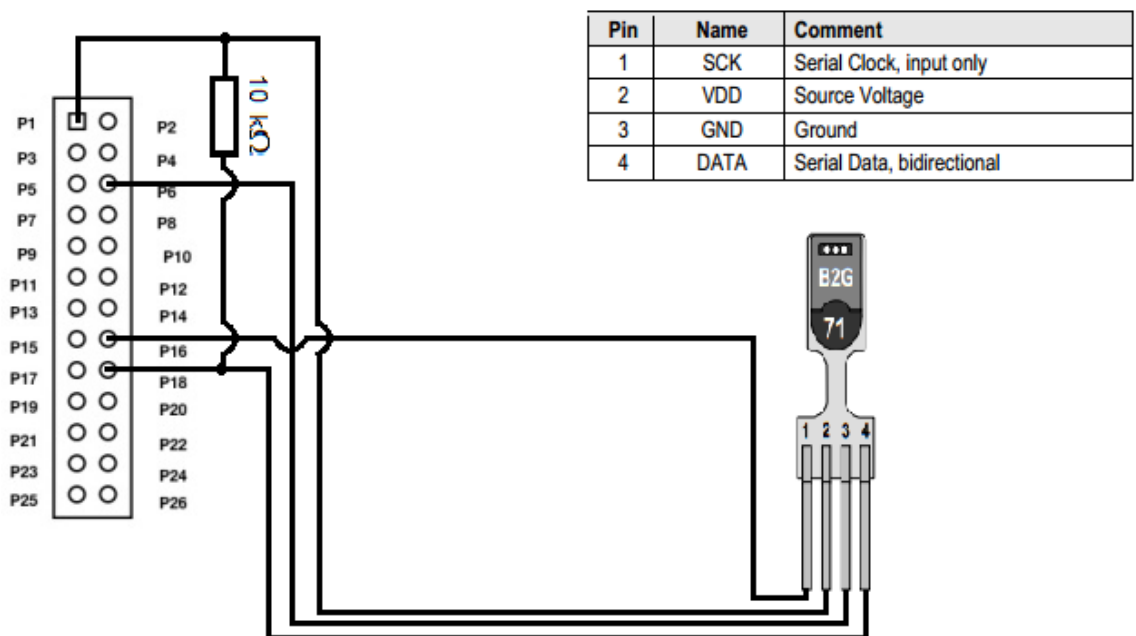
SHT71 on digitaalinen, I2C-väylää hyödyntävä, täysin kalibroitu ja pienivirtainen anturi. Anturi on erittäin helppo kytkeä, sillä se sisältää vain neljä pinniä: SDA, SCL, maa ja käyttöjännite, joka voi olla välillä 2,4 V–5 V (Suositus on 3,3 V, jota tässäkin työssä käytetään). [26, s. 1.]

### 6.1.1 Anturin kytkentä

Anturin kytkeminen Raspberry Pi:lle on hyvin yksinkertaista, sillä anturissa on vain neljä pinniä. Anturi ei vaadi myöskään kuin yhden ulkoisen komponentin, joka on käyttöjännite- ja datalinjan välille tuleva  $10\text{ k}\Omega$  vastus, jolla mikrokontrolleri voi ajaa linjan ylös. [26, s. 4.]

Anturia ei kuitenkaan kytketä suoraan Raspberry Pi:n GPIO-liittimen I2C-pinneihin, vaan ihan tavallisiin GPIO-pinneihin. Tällä tavalla toteutettu I2C-tiedonsiirto on hieman haasteellisempi kuin suoraan Raspberry Pi:n GPIO-liittimen I2C-pinnien kautta suoritettu tiedonsiirto.

Anturin 1-pinni eli SCK kytketään Raspberry Pi:n GPIO-pinniin P18. Anturin VDD-pinni kytketään GPIO-pinniin P1, josta anturi saa  $3,3\text{ V}$  käyttöjännitteen. GND eli maapinni kytketään pinniin P6. Datapinni eli anturin 4. pinni kytketään sekä suoraan GPIO-pinniin P18 että  $10\text{ k}\Omega$ :n vastuksen kautta  $3,3\text{ V}$ :n käyttöjännitepinniin P1. Kytkentä on nähtävissä kokonaisuudessaan kuvassa 27.



Kuva 27. SHT71-anturin kytkeminen Raspberry Pi:n GPIO-liittimeen [26, s. 4]

### 6.1.2 Lämpötilan ja kosteuden lukeminen anturilta

Lämpötilan ja kosteuden mittauksen ohjelmakoodi on toteutettu niin, että ohjelmakoodi olisi mahdollisimman selkeästi luettavaa. Kaikki eri anturilla suoritettavat toimenpiteet ja anturille lähetettävät käskyt on toteutettu omissa funktioissaan.

Lämpötilan ja kosteuden lukeminen aloitetaan main-funktiossa while-loopin sisässä funktiolla `MeasureTempAndHum`, joka palauttaa luetun mittausdatan tietueen osoitteen. Funktiolle ei välitetä mitään parametreja. `MeasureTempAndHum`-funktion alussa on heti 20 ms viive, koska anturi vaatii vähintään 11 ms viiveen päästäkseen sleep-tilaan, joka vaaditaan mittauksen aloittamiseen.

Seuraavaksi alustetaan data- ja kellopinnit `SHT7x_InitPins`-funktioilla. Funktiolla ei ole palautusarvoa, eikä sille välitetä mitään parametreja. Funktiossa asetetaan kellolinja ulostuloksi ja datalinja sisääntuloksi.

Anturia ei ole pakko resetoida, mutta se on suositeltavaa seuraavaksi. Funktio `SHT7x_Reset` funktio resetoi anturin. Se ei palauta mitään, eikä sille välitetä mitään parametreja. `SHT7x_Reset`-funktiossa asetetaan ensin datalinja ykköstitilaan, jonka jälkeen asetetaan kellolinja yhdeksän kertaa ykköstitilasta nollatilaan. Tätä seuraa `SHT7x_Transmission_Start`-funktio, jossa annetaan aloituskäsky asettamalla data- ja kellolinjoja eri tilaan. Tämän jälkeen resetoidaan crc-laskenta funktiolla `SHT7x_crc`, joka saa parametrina `SHT7x_Mirrorbyte`-funktion palauttaman arvon. `SHT7x_Mirrorbyte`-funktio kääntää sille parametrina annetun muuttujan bittien järjestyksen. Ohjelma palaa takaisin resetointi-funktioon, ja lähettää lopulta `SHT7x_Sendbyte`-funktioilla resetoinnin vaatiman heksadesimaalisen luvun, `0x1E`, mikä saa aikaan lopulta anturin nollauksen.

`MeasureTempAndHum`-funktiossa kutsutaan seuraavaksi funktiota `SHT7x_Measure_Start`, jolle välitetään parametrina `0x03`, joka on lämpötilan mittauksen aloittava heksadesimaalinen arvo. `SHT7x_Measure_Start`-funktio alkaa `SHT7x_Transmission_Start`-funktioilla, jota seuraa `SHT7x_Sendbyte`-funktio, joka palauttaa vahvistusbitin, ja saa parametrina lämpötilanmittauksen aloittavan heksadesimaalisen luvun, `0x03`.

Seuraavaksi kutsutaan `SHT7x_Get_Measure_Value`-funktioita, jolla luetaan lämpötiladata anturilta. Funktiolle välitetään parametrina union-tietorakenteen lämpötilan jäsenmuuttuja. Funktion sisällä luetaan kaksi tavua lämpötiladataa kutsumalla `SHT7x_Readbyte`-funktioita,

jonka jälkeen tehdään crc-laskenta SHT7x\_Crc\_Check-funktiolla, ja lopuksi kutsutaan SHT7x\_Mirrorbyte-funktiota, joka kääntää bittien järjestyksen ja palauttaa lämpötilan arvon.

Kosteuden mittaukselle on ihan sama periaate kuin lämpötilankin mittaukselle. Sillä erolla, että kutsuttaessa SHT7x\_Measure\_Start-funktiota funktiolle välitetäänkin parametriksi kosteudenmittauksen aloittavan heksadesimaalisen luvun, eli 0x05. SHT7x\_Get\_Measure\_Value-funktiota kutsuttaessa funktiolle välitetään parametriksi union-tietorakenteen kosteuden jäsenmuuttuja.

Kumpikin mittausarvo on kokonaislukutyypinen, joten arvoille täytyy tehdä castaus float-, eli liukulukutietotyyppiseksi. Mittausarvot ovat vielä raaka-arvoja, joten niille lasketaan oikeat arvot SHT7x\_Calc-funktiolla, jolle välitetään parametrina union-tietorakenteen kummankin jäsenmuuttujan osoite.

## 6.2 Ilmanpaineen mittaus

Ilmanpaineen mittaukseen päätettiin käyttää Freescalen Xtrinsic MPL3115A2 -paineanturia. MPL3115A2-anturi on I2C-väylää hyödyntävä vähävirtainen ja erittäin tarkka pintaliitosmikropiiri. Anturista oli saatavilla ”breakout board” eli n. sormenpään kokoinen piirikortti, johon on integroitu itse anturi sekä anturin tarvitsemat passiiviset komponentit.

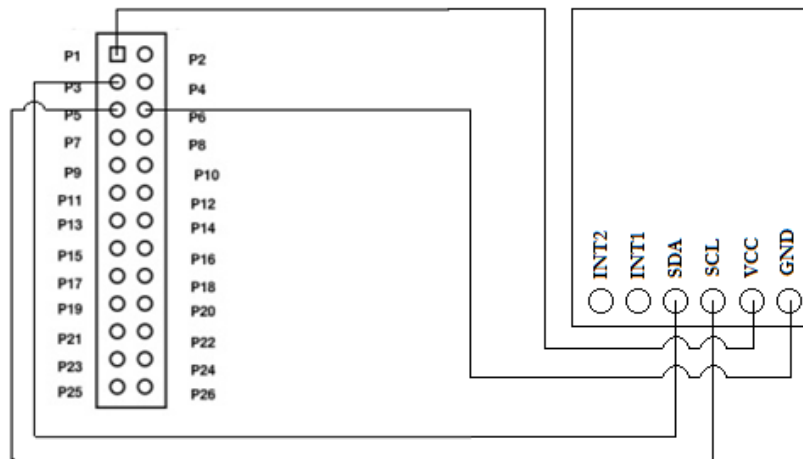
Breakout board vähentää anturin pinnien määrää kahdella kappaleella, eli pinnejä on yhteensä 6 kappaletta. SCL-, SDA-, VCC-, GND-pinnit ovat käytössä tässä työssä. Lisäksi on kaksi ulkoisille keskeytyksille tarkoitettua INT1- ja INT2-pinniä, joita ei tässä työssä tarvita.

### 6.2.1 Anturin kytkentä

Paineanturi kytketään suoraan Raspberry Pi:n GPIO-liittimen I2C-pinneihin, jolloin ohjelmakoodista tulee hieman yksinkertaisempi kuin, jos anturi kytkettäisiin tavallisiin GPIO-pinneihin.

Anturin VCC eli käyttöjännite kytketään Raspberry Pi:n GPIO-liittimen P1-pinniin. Käyttöjännitteen tulee olla välillä 1,95–3,6 V. GND-pinni kytketään GPIO-liittimen P6-pinniin.

SDA- ja SCL-pinnit kytketään GPIO-liittimen vastaaviin pinneihin eli SDA-pinniin P3 ja SCL-pinniin P5. MPL3115A2-anturin kytkentä Raspberry Pi:hin on nähtävissä kuvassa 28.



Kuva 28. MPL3115A2-anturin kytkeminen Raspberry Pi:n GPIO-liittimeen [27, s. 3] [27, s. 4]

### 6.2.2 Ilmanpaineen lukeminen anturilta

Ilmanpaineen mittauksessa on käytetty hyväksi bcm2835-kirjaston sisältämiä valmiita I2C-tiedonsiirtoon tarkoitettuja funktioita. Käyttämällä bcm2835-kirjaston valmiita funktioita saadaan ohjelmakoodista huomattavasti lyhyempi ja yksinkertaisempi.

Ilmanpaineen mittauksenkin ohjelmakoodi on pyritty pitämään mahdollisimman selkeästi hahmotettavana ja yksinkertaisena. Kaikki anturille tehtävät rekisterimuokkaukset on toteutettu omissa funktioissaan, joita sitten kutsutaan tarvittaessa.

Mittaus aloitetaan kutsumalla main-funktiossa ilmanpaineenmittausfunktiota `Get_Pressure_Value`. Funktiolle välitetään parametreina taulukko, johon mitattu paine tallennetaan, sekä taulukon pituus. Funktio on while-loopin sisässä, jotta mittaus olisi jatkuvaa.

`Get_Pressure_Value`-funktion alussa määritellään kolme tavua pitkä taulukko `pressure_data_register`, johon sijoitetaan anturin kolmen tavun paineenmittauksen rekisteriarvot: `0x01`, `0x02` sekä `0x03`.



Seuraavana on kolme funktiota, jolla I2C-väylä saadaan alustettua. Ensimmäinen funktio on `bcm2835_i2c_begin`, jolle ei välitetä mitään parametreja. Seuraava funktio eli `bcm2835_i2c_setSlaveAddress` asettaa sen laitteen osoitteen, jonka kanssa halutaan keskustella. Funktiolle välitetään parametriksi paineanturin laiteosoite, `0x60`. Kolmannella alustus-funktiolla asetetaan väylän kellotaajuus halutuksi. `Bcm2835_i2c_setClockDividerille` välitetään parametriksi `BCM2835_I2C_CLOCK_DIVIDER_148`, jolloin kellotaajuudeksi saadaan `1,689 Mhz`.

Väylän alustuksen jälkeen seuraa paineanturin rekisterien muokkausta haluttujen asetusten aikaansaamiseksi. Anturi on oletuksena odotustilassa, mikä mahdollistaa anturin `CTRL_REG1`-rekisterin muokkauksen. On kuitenkin hyvä varmistaa, että anturi on odotustilassa. `SetModeStandby`-funktiolla nollataan `CTRL_REG1`-rekisterin ensimmäinen eli `SBYB`-bitti, joka asettaa anturin odotustilaan.

Anturi saadaan painemittausmoodiin `setModeBarometer`-funktiolla, joka nollaa `CTRL_REG1`-rekisterin viimeisen eli `ALT`-bitin. `SetOSR128`-funktiolla asetetaan `CTRL_REG1`-rekisterin arvoksi `0x38`, jolloin saadaan anturi ottamaan mittausnäytteitä `512 ms` välein. `EnableEventFlags`-funktio asettaa `PT_DATA_CFG`-rekisterin arvoksi `0x07`, joka ottaa käyttöön paineenmittausliput.

Lopuksi asetetaan anturi aktiiviseen tilaan `setModeActive`-funktiolla, joka asettaa `CTRL_REG1`-rekisterin ensimmäisen eli `SBYB`-bitin ykköstitilaan. Kun anturin on aktiivisessa tilassa, voidaan sillä aloittaa lukemaan dataa. `Bcm2835_i2c_read_register_rs`-funktio lukee anturilta painedatan `pressure_buf`-taulukkoon.

Ilmanpaineen mittaus ei lopullisessa versiossa toimi. Ilmanpaineen mittaus toimi kyllä jonkin aikaa, kunnes jostakin syystä se lakkasi toimimasta. Toimimattomuudelle ei ole löytynyt mitään syytä.

## 6.3 Bluetooth-yhteys

### 6.3.1 Bluetooth USB-adapterin käyttöönotto Raspberry Pi:ssä

Raspberry Pi ei itsessään sisällä Bluetoothia, joten se vaatii erillisen USB-porttiin liitettävän Bluetooth USB-adapterin sekä sille soveltuvan laiteajurin.

Päätettiin käyttää B-Speechin Bluetooth USB-adapteria. Adapteri on euron kolikon kokoinen, ja se tukee Bluetoothin versiota 2. Sen tiedonsiirtonopeus on maksimissaan 3 Mbit/s.

Ennen kuin Raspberry Pi:lle ruvetaan asentamaan Bluetooth USB-adapterille tarkoitettua laiteajuria, on hyvä päivittää viimeisimmät ohjelmistopakettit Raspberry Pi:lle. Tämä tapahtuu kirjoittamalla komentoriville komento ”sudo apt-get update && sudo apt-get upgrade”. Ensimmäinen komento hakee tarvittavat päivitykset, ja jälkimmäinen asentaa ne laitteelle. Jos haluaa, niin voi ajaa vielä komennon ”sudo apt-get autoremove”, jolla poistetaan kaikki tarpeettomat ohjelmistopakettit.

Seuraavaksi asennetaan laiteajuri Bluetooth USB-adapterille komennolla ”sudo apt-get install bluetooth bluez-utils blueman”.

Raspberry Pi kannattaa tässä vaiheessa käynnistää uudelleen, jonka jälkeen on hyvä kokeilla onnistuiko laiteajureiden asennus. Komennolla ”lsusb” saadaan näkyviin lista kaikista laitteeseen kytketyistä USB-laitteista. Listassa pitäisi näkyä nimi: **Bus 001 Device 004: ID 0a12:0001 Cambridge Silicon Radio, Ltd Bluetooth Dongle (HCI mode)**.

On mahdollista myös kokeilla, onko Bluetooth käynnissä kirjoittamalla komentoriville komento ”/etc/init.d/bluetooth status”. Komennon jälkeen ruutuun pitäisi ilmestyä seuraavanlainen viesti: **[ ok ] bluetooth is running**.

Komennolla ”hcitool scan” on mahdollista skannata adapterin kantama-alueen sisällä olevia laitteita.

### 6.3.2 Bluetooth-tiedonsiirto

Bluetooth-ajureiden asentaminen ei vielä asenna Raspberry Pi:lle Bluetooth C-ohjelmoinnin vaatimia .header-tiedostoja, joten ne täytyy asentaa kirjoittamalla Raspberry Pi:n komentoriville komento ”sudo apt-get install libbluetooth-dev”. Seuraavaksi mennään kansioon /usr/lib/arm-linux-gnueabi/hf, ja kopioidaan kirjastotiedostot libbluetooth.a ja libbluetooth.so. Tiedostot täytyy viedä Eclipsen kääntäjän lib-kansioon, joka sijaitsee polussa home/rpi/tools/arm-bcm2708/gcc-linaro-arm-linux-gnueabi/hf-raspbian/arm-linux-gnueabi/hf/lib. Lisätään vielä Eclipsen linkkerille lippu -lbluetooth. Lopuksi mennään polkuun /usr/include, ja kopioidaan bluetooth-kansio, joka sisältää tarvittavat .header-tiedostot Bluetooth-C-ohjelmointia varten. Kopioitu kansio viedään Eclipsen kääntäjän include-kansioon, joka on polussa /home/yourusername/rpi/arm-bcm2708/gcc-linaro-arm-linux-gnueabi/hf-raspbian/arm-linux-gnueabi/hf/libc/usr/include.

Jotta Bluetooth-tiedonsiirto toimisi Raspberry Pi -minitietokoneiden välillä, täytyy laitteet parittaa keskenään, sekä näkyä toisilleen ns. ”trusted”-tilassa. Bluetooth-laitteesta saadaan näkyvä kirjoittamalla Raspberry Pi ”B”-n komentoriville komento ”sudo hciconfig hci0 piscan”. Tämän jälkeen skannataan lähellä olevat Bluetooth-laitteet kirjoittamalla Raspberry Pi ”A”-n komentoriville komento ”sudo hcitool scan”. Tällä saadaan näkyviin Raspberry Pi ”B”-n Bluetooth-laiteosoite, joka on muotoa XX:XX:XX:XX:XX:XX. Ajetaan parituskomento ”sudo bluez-simple-agent” Raspberry Pi ”B”-llä, jonka jälkeen Raspberry Pi ”A”-lla ajetaan komento ”sudo bluez-simple-agent hci0 ’Raspberry Pi ”B”-n -laiteosoite’ ”. Annetaan PIN-koodi, joka on oletuksena 0000, ja annetaan sama PIN-koodi myös Raspberry Pi ”B”-n puolella. Lopuksi tehdään Bluetooth-laitteesta ”trusted” kirjoittamalla Raspberry Pi ”A”-n komentoriville komento ”sudo bluez-test-device trusted ’Raspberry Pi ”B”-n -laiteosoite’ ”.

Bluetooth-ohjelmointi perustuu samaan socket-periaatteeseen kuin Internet-ohjelmointikin. Bluetooth-ohjelmoinnin kaksi yleisintä socket-protokollaa ovat RFCOMM (Radio Frequency Communication) ja L2CAP, joista jälkimmäinen on käytössä tässä insinööriyössä. L2CAP-socket mahdollistaa muun muassa muutaman lisäominaisuuden kontrolloimisen.

Kuten Internet-ohjelmoinnissakin, on Bluetooth L2CAP-socket-ohjelmoinnissa sekä palvelin- että asiakasyhteydet. Tässä työssä palvelimen roolia hoitaa Raspberry Pi, joka vastaanottaa mittausdataa, ja asiakkaan roolia hoitaa Raspberry Pi, joka lukee anturidataa.

Asiakasyhteyden puolella Bluetooth-yhdistäminen ja Bluetooth-lähetys on kumpikin toteutettu omassa funktiossaan. Bluetooth-yhteyden muodostamiseen riittää yksi kerta, joten sitä funktiota kutsutaan vain kerran. Bluetooth-lähetysfunktio laitetaan while-loopin sisään, jotta lähetyksestä saataisiin jatkuva-aikainen.

Bluetooth-yhdistämisfunktio on nimeltä `L2CAP_client_connection`, joka palauttaa integer eli kokonaislukutietotyyppiä olevan osoitteen socketin luomisen yhteydessä saadusta arvosta. Tätä arvoa tarvitaan parametriksi lähetysfunktioille. Yhdistämisfunktioille välitetään parametrina Raspberry Pi:n Bluetooth-adapterin laiteosoite, joka on `00:15:83:32:7A:3F`. `L2CAP_client_connection`-funktio pitää sisällään muun muassa funktion, jolla varataan socketti, yhdistämisparametrien asetukset sekä funktion, jolla varsinainen yhdistäminen tehdään.

Bluetooth-lähetysfunktio on nimeltä `L2CAP_client`. Sillä ei ole palautusarvoa, mutta sille välitetään neljä eri parametria: mitatun lämpötilan, kosteuden ja paineen muuttujien osoitteet sekä socketin arvo. `L2CAP_client`-funktio ei pidä sisällään juuri muuta kuin varsinaisen lähetysfunktion, jolla mittausdata lähetetään eteenpäin.

Palvelinyhteyden puolella on käytetty ihan samaa periaatetta kuin asiakasyhteydenkin puolella, eli Bluetooth-yhdistäminen ja Bluetooth-vastaanotto on toteutettu kumpikin omassa funktiossaan, koska yhdistämiseen riittää yksi kerta, mutta dataa vastaanotetaan koko ajan.

Palvelinyhteyden puolella Bluetooth-yhdistämisfunktio on nimeltä `L2CAP_server_connection`, joka palauttaa kokonaislukutietotyyppiä olevan osoitteen yhteyden hyväksymisen yhteydessä saadusta arvosta, jota tarvitaan parametriksi vastaanottofunktioille. `L2CAP_server_connection`-funktioille ei välitetä mitään parametreja. `L2CAP_server_connection`-funktio pitää sisällään muun muassa socketin varausfunktion, yhdistämisfunktion, socketin kuuntelufunktion sekä Bluetooth-yhteyden hyväksymisfunktion.

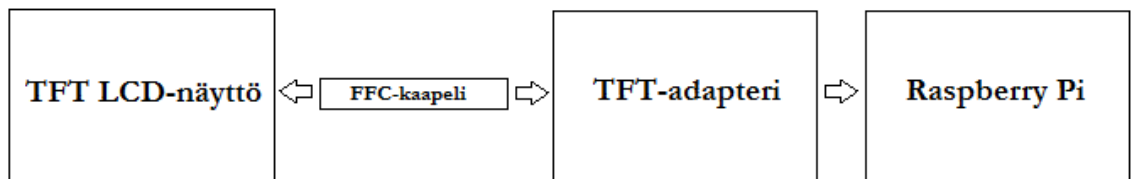
Bluetooth-vastaanottofunktio on nimeltä `L2CAP_server`. Se palauttaa mittausdatan tietueen osoitteen, ja sille välitetään parametrina yhteyden hyväksymisen yhteydessä saatu arvo. `L2CAP_server`-funktio ei pidä sisällään muuta kuin mittausdatan varsinaisen vastaanottamiseen tarkoitetun vastaanottofunktion tarvittavineen muuttujineen.

## 6.4 LCD-näyttö

LCD-näyttönä päätettiin käyttää saksalaisen Admatecin valmistamaa 3,5'', LED-taustavalaistua (Light-Emitting Diode), TFT (Thin-Film-Transistor) LCD-näyttöä. Näytön resoluutio on 320x240 pikseliä, ja näytönohjaimena toimii RA8870. Näyttö kytketään mukana tulevalla apukortilla suoraan Raspberry Pi:n GPIO-porttiin, ja se käyttää SPI-väylää tiedonsiirtoon.

### 6.4.1 LCD-näytön kytkeminen

Admatecin LCD-näyttöpaketti sisältää kolme osaa: itse TFT LCD-näytön, TFT-adapterin ja FFC-lattakaapelin (Flat Flex Cable). FFC-lattakaapelin toinen pää kytketään TFT LCD-näytön FFC-lattakaapelin liittimeen, ja FFC-lattakaapelin toinen pää TFT-adapterin FFC-lattakaapelin liittimeen. Lopuksi kytketään TFT-adapteri suoraan Raspberry Pi:n GPIO-porttiin. LCD-näytön kytkeminen Raspberry Pi:hin on nähtävissä kuvassa 29.



Kuva 29. LCD-näytön kytkeminen Raspberry Pi:hin

### 6.4.2 LCD-näytön alustaminen C-ohjelmointikielellä

LCD-näytölle on valmiit C-kieliset .c- ja .h-tiedostot näytön alustamista varten. Ne saadaan ladattua kirjoittamalla komentoriville komento ”sudo wget <http://admatec.de/sites/default/files/downloads/C-Berry.tar.gz>”. Paketti puretaan komennolla ”tar zxvf C-Berry.tar.gz”. Purettu kansio sisältää näytön alustamiseen tarvittavat .c- ja

.h-tiedostot, jotka voidaan suoraan lisätä omaan projektiin. Jotta näyttö toimisi, täytyy bcm2835-kirjasto olla asennettuna.

LCD-näyttö saadaan alustettua kutsumalla kolmea eri funktiota: `TFT_init_board`, `TFT_hard_reset` sekä `RAIO_init`. `RAIO_clear_screen`-funktiolla saadaan LCD-näyttö tyhjennettyä.

LCD-näytölle ladatut ohjelmakirjastot sisältävät valmiin funktion, jolla LCD-näytölle saadaan tulostettua tekstiä. Funktio on nimeltä `example_WriteText`, jolle välitetään parametrina muuttujatyypiltään unsigned char- eli etumerkitön merkki. Funktion sisällä voi käydä itse muokkaamassa tulostettavan tekstin kokoa ja väriä.

Mittausdata on liukulukumuuttujatyyppejä, ja `example_WriteText`-funktiolle ei voi välittää parametriksi liukulukumuuttujaa, joten liukuluku täytyy sijoittaa ensin etumerkittömään merkkitaulukkoon `sprintf`-funktiolla. Esimerkkinä mitatulle lämpötilalle funktio näyttää kokonaisuudessaan tältä: `sprintf((char*)buf1, "%0,1f C", m_Data->temp)`. Funktiolle välitetään ensimmäiseksi parametriksi `buf1`-taulukko, joka on muuttujatyypiltään etumerkitön merkki. Koska funktiolle välitettävän ensimmäisen parametrin on oltava muuttujatyypiltään merkki, täytyy `buf1`-taulukolle tehdä castaus (char\*)- eli merkkimuuttujatyypiksi. Toinen funktion parametri on tulostettavan merkkijonon muoto, eli tässä tapauksessa yhden desimaalin liukuluku. Viimeiseksi parametriksi välitetään mittausdatan tietueen lämpötilamuuttujan osoitinmuuttuja.

## 6.5 Käyttöliittymäsovellus

Käyttöliittymäsovelluksella ei ollut juuri muita vaatimuksia kuin, että ohjelmakoodin toteutus täytyi dokumentoida niin, että sitä olisi mahdollisimman helppo jatkossa muokata.

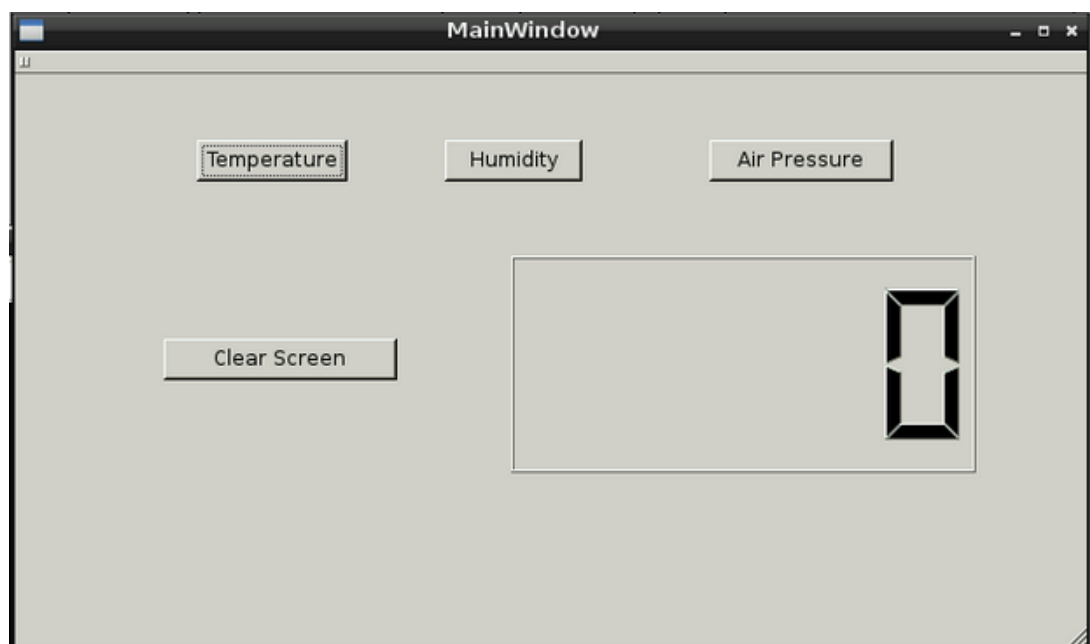
Käyttöliittymän kehittäminen tapahtui Qt Creatorilla sen suosion, ja suosion aikaansaaman dokumentoinnin/ohjeistuksen tarjonnan vuoksi. Qt Creatorilla ristiinkääntäminen toiselle laitearkkitehtuurille on melko monimutkaista. Se olisi luultavasti tuottanut huomattavan paljon ongelmia, joten päädyin kehittämään ohjelmakoodia suoraan Raspberry Pi:llä, vaikka ohjelmakoodin kääntäminen viekin melko paljon aikaa siinä. Qt Creatorilla käyttöliittymän kehittäminen tapahtuu C++-ohjelmointikieltä käyttäen.

## Käyttöliittymän kehittäminen Qt Creatorilla

Kun Qt Creatorilla aloitettiin käyttöliittymän kehittäminen, lisättiin jo Eclipsellä valmiiksi luodut ohjelmakoodin .c- ja .h-tiedostot ja LCD-näytön alustamiseen tarvittavat valmiit .c- ja .h-tiedostot luotuun projektiin. Ainut muutos mitä piti tehdä, oli Bluetooth-vastaanottofunktion laittaminen omaan luokkaan. Bluetooth-vastaanottofunktion laittaminen omaan luokkaan täytyi tehdä, jotta data saataisiin siirrettyä painike-funktiolle.

Itse käyttöliittymän ulkoasun luonti on hyvin yksinkertaista. Luodaan Qt Gui Application -projekti, ja valitaan .ui-tiedosto. Tässä .ui-tiedostossa on mahdollista luoda käyttöliittymäsovelluksen ulkoasu. Vasemmassa reunassa on erilaisia widgettejä, joita voidaan lisätä käyttöliittymän ulkoasun luontiin tarkoitettuun kohtaan.

Tämän insinööriyön käyttöliittymäikkuna on pidetty ulkoasultaan melko yksinkertaisena sisältäen vain viisi eri widgettiä: neljä painike-widgettiä sekä yhden LCD-näyttö-widgetin. Temperature-, Humidity- ja Air Pressure -painikkeista tulostuu kyseisen painikkeen mittausdatan arvo LCD-näyttö-widgettiin sekä Raspberry Pi:n GPIO-porttiin kytkettyyn LCD-näyttöön. Clear Screen -painikkeesta painettaessa nollautuu sekä LCD-näyttö-widgetti että itse fyysinen LCD-näyttö. Järjestelmälle suunniteltu käyttöliittymäsovellus on nähtävissä kuvassa 30.



Kuva 30. Järjestelmälle toteutettu käyttöliittymäsovellus

Kun widgetiä klikkaa sen luomisvaiheessa hiiren oikealla näppäimellä ja valitsee kohdan ”Go to slot...” ja edelleen ”clicked()”, Qt Creator luo automaattisesti kyseiselle widgetille valmiin funktion, jonka sisään voi kirjoittaa halutun toiminnallisuuden. Kyseinen clicked()-valinta tarkoittaa sitä, että kun nappia painetaan, suoritetaan funktion sisällä oleva ohjelmakoodi.

Käyttöliittymää ei ehditty lopulta saada annetun aikarajan puitteissa valmiiksi. Ongelmaksi muodostui Bluetoothilta tulevan datan saaminen painike-widgettien funktioiden sisään. Bluetoothilta tulevan datan vastaanottaminen onnistui muualla ohjelman rungossa, mutta painike-widgetin sisään dataa ei saatu tuotua.



## 7 TYÖN/TULOSTEN ANALYSOINTI

Työn tuloksena saatiin lähes toimiva mittausjärjestelmä. Mittausjärjestelmä mittaa Raspberry Pi:llä lämpötilaa sekä ilmankosteutta ja lähettää mittausdataa jatkuvana datavirtana Bluetoothiin avulla eteenpäin toiselle Raspberry Pi:lle, jossa mittausdata tulostuu LCD-näytölle. Paineenmittaus toimi hetken, jonka jälkeen tuntemattomasta syystä lakkasi toimimasta, ja käyttöliittymäsovellusta ei saatu täysin toimimaan.

Mittausjärjestelmälle suunniteltu ohjelmakoodi on pyritty toteuttamaan siten, että ohjelman hahmotus olisi mahdollisimman helppoa. Kaikki ohjelmakoodissa olevat osiot ovat omissa c-tiedostoissaan, ja näiden tiedostojen sisällä on vielä kaikki osion sisältämät funktiot. Funktioille on annettu myös nimet, jotka kuvaavat funktion sisältämää toimintaa. Ohjelmakoodi sisältää myös defines-tiedoston, jossa on kasattu esikäntäjän otsikkotiedostot, vakiot, makrot, tietueiden määrittelyt ja funktioiden esittelyt.

Työn teon aikana tuli vastaan yllättävän paljon ongelmia etenkin kehitysympäristöjen kanssa. Suurimmat ongelmat olivat Qt Creator -kehitysympäristön kanssa, sillä kehitysympäristön käyttämisen joutui opettelemaan täysin alusta alkaen. Ohjelmakoodin toteuttamisessa suurimmat ongelmat olivat käyttöliittymäsovelluksen kanssa, mistä johtuen se ei työn lopullisessa versiossa toimi.

Koska työn tekoon annettu aika loppui kesken, lopullisessa versiossa paineenmittaus ei toimi. Paineenmittaus ehti kuitenkin toimia jonkin aikaa ennen kuin se lopetti toimimisen eikä syytä paineenmittauksen toimimattomuudelle kuitenkaan keretty lopulta löytää. Ohjelmakoodissa vian ei pitäisi olla, sillä ohjelmakoodin versioon, jolla paineenmittaus alun perin toimi, ei tullut mitään muutoksia. Anturin kytkennätkin oli tarkastettu useasti, ja ne olivat kunnossa. Itse anturia, joka todennäköisesti oli ongelman aiheuttaja, ei ehditty erinäisistä syistä lopulta tarkistamaan tai vaihtamaan uuteen.

Käyttöliittymäsovelluksessa ongelmana oli saada Bluetooth-palvelimelta tuleva mittausdata toimimaan painike-widgettien sisällä. Jostain syystä muualla ohjelman rungossa Bluetooth-palvelimelta tuleva mittausdata saatiin toimimaan. Todennäköisesti Measurement Data -tietue pitäisi laittaa omaan luokkansa, ja tehdä luokan sisässä jokaiselle painike-widgetille oma Get-funktio, jolla mittausdata vastaanotettaisiin. Measurement Data -tietue oli se mihin Bluetooth-vastaanottofunktio palautti vastaanotetun mittausdatan. Insinööriyössä tuli kui-

tenkin aikaraja vastaan, joten emme lopulta ehtineet kokeilla kyseistä metodia emmekä saaneet käyttöösi toimimaan.

## 8 YHTEENVETO

Tässä insinööriyössä oli tavoitteena toteuttaa Kajaanin ammattikorkeakoululle langaton mittausjärjestelmä, jossa Raspberry Pi -minitietokoneella mitataan lämpötilaa, ilmankosteutta sekä ilmanpainetta. Mittausdata piti lähettää langattomasti Bluetoothin avulla toisella Raspberry Pi:lle, jossa mittausdataa voitaisiin tarkastella visuaalisesti Raspberry Pi:n GPIO-porttiin kytketystä LCD-näytöstä sekä järjestelmälle toteutettavasta käyttöliittymäsovelluksesta. Kaikkia työlle asetettuja tavoitteita ei lopulta saatu kuitenkaan toteutettua, sillä lopullisessa versiossa paineenmittaus ja käyttöliittymäsovellus eivät toimineet.

Henkilökohtainen tavoitteeni oli kehittyä ohjelmoijana mahdollisimman paljon. Siksi olinkin antanut insinööriyöni aiheelle yhden tärkeän kriteerin: työn tuli sisältää paljon ohjelmointia. Insinööriyöni olikin pääosin ohjelmointiin keskittyvä työ, ja mielestäni kehityin ohjelmoijana erittäin paljon, vaikka lopullisessa ohjelmakoodin versiossa ei kaikki ihan täysin toiminutkaan. Työ täytti onneksi ainakin henkilökohtaisen tavoitteen, vaikkei täyttäneenkään täysin työlle asetettuja tavoitteita.

Insinööriyön tekeminen on ollut mielenkiintoinen ja erittäin opettavainen prosessi. Se on opettanut erittäin paljon etenkin ohjelmointia sekä myös dokumentointia ja projektiluonteista työskentelytapaa. Työn tekeminen opetti pitkäjänteisyyttä sekä ongelmanratkaisukykyä, sillä erinäisten ongelmien tullessa eteen jouduin etsimään ongelmaan yleensä itse ratkaisun.

## LÄHTEET

1. What is a Raspberry Pi? [WWW-dokumentti].  
<<http://www.raspberrypi.org/help/what-is-a-raspberry-pi/>>. (Luettu 23.6.2014)
2. About Us. [WWW-dokumentti].  
<<http://www.raspberrypi.org/about/>>. (Luettu 23.6.2014)
3. element14. RaspberryPi. [WWW-dokumentti].  
<<http://downloads.element14.com/raspberryPi1.html>>. (Luettu 24.6.2014)
4. Raspberry Pi. [WWW-dokumentti].  
<[http://en.wikipedia.org/wiki/Raspberry\\_Pi](http://en.wikipedia.org/wiki/Raspberry_Pi)>. (Luettu 24.6.2014)
5. eLinux. RPi Low-Level peripherals. [WWW-dokumentti].  
<[http://elinux.org/RPi\\_Low-level\\_peripherals](http://elinux.org/RPi_Low-level_peripherals)>. (Luettu 24.6.2014)
6. adafruit. The GPIO Connector. [WWW-dokumentti].  
<<https://learn.adafruit.com/adafruits-raspberry-pi-lesson-4-gpio-setup/the-gpio-connector>>. (Luettu 25.6.2014)
7. Wentk Richard, 2014, Teach Yourself VISUALLY Raspberry Pi, Luettu 25.6.2014,  
<http://www.kamk.fi>, Nelli-portaali, Ebrary
8. Welcome to Raspbian. [WWW-dokumentti].  
<<http://www.raspbian.org/>>. (Luettu 26.6.2014)
9. About Raspbian. [WWW-dokumentti].  
<<http://www.raspbian.org/RaspbianAbout>>. (Luettu 26.6.2014)
10. Raspbian FAQ. [WWW-dokumentti].  
<<http://www.raspbian.org/RaspbianFAQ>>. (Luettu 27.6.2014)
11. I2C-Bus: What's that? [WWW-dokumentti].  
<<http://www.i2c-bus.org/>>. (Luettu 27.6.2014)
12. I2C. [WWW-dokumentti].  
<<http://en.wikipedia.org/wiki/I%C2%B2C>>. (Luettu 27.6.2014)
13. Using the I2C Bus. [WWW-dokumentti].  
<[http://www.robot-electronics.co.uk/acatalog/I2C\\_Tutorial.html](http://www.robot-electronics.co.uk/acatalog/I2C_Tutorial.html)>. (Luettu 2.7.2014)
14. Introduction to I2C. [WWW-dokumentti].  
<<http://www.embedded.com/electronics-blogs/beginner-s-corner/4023816/Introduction-to-I2C>>. (Luettu 2.7.2014)

15. I2C Info - I2C Bus, Interface and Protocol. [WWW-dokumentti].  
<<http://i2c.info/>>. (Luettu 2.7.2014)
16. A Single master I2C tutorial. [WWW-dokumentti].  
<<http://www.best-microcontroller-projects.com/i2c-tutorial.html>>. (Luettu 2.7.2014).
17. An I2C Network Protocol for Environmental Monitoring. [PDF-dokumentti].  
<<http://ww1.microchip.com/downloads/en/AppNotes/00736a.pdf>>. (Luettu 3.7.2014)
18. Granlund Kaj, Tietoliikenne, Jyväskylä: WSOYpro/Docendo, 2007, 1. painos, ISBN 978-951-0-32821-7.
19. Bluetooth. [WWW-dokumentti].  
<<http://en.wikipedia.org/wiki/Bluetooth>>. (Luettu 5.7.2014)
20. Frequency Modulation. [WWW-dokumentti]  
<[http://en.wikipedia.org/wiki/Frequency\\_modulation](http://en.wikipedia.org/wiki/Frequency_modulation)>. (Luettu 7.7.2014)
21. Optimization of bluetooth frame format for efficient performance. [PDF-dokumentti]. <<http://www.jpier.org/PIERM/pierm01/09.08012302.pdf>>. (Luettu 9.7.2014)
22. Granlund Kaj, Langaton tiedonsiirto, Jyväskylä: Docendo, 2001, 1. painos, ISBN 951-846-091-4
23. About. [WWW-dokumentti].  
<<http://bnap.opensecurityresearch.com/readme.html>>. (Luettu 22.7.2014)
24. Secure Shell (SSH). [WWW-dokumentti].  
<<http://searchsecurity.techtarget.com/definition/Secure-Shell>>. (Luettu 3.7.2014)
25. Development Environment for the Raspberry Pi using a Cross Compiling Toolchain and Eclipse. [WWW-dokumentti].  
< <http://hertaville.com/2012/09/28/development-environment-raspberry-pi-cross-compiler/>>. (Luettu 4.7.2014)
26. Datasheet SHT7x (SHT71, SHT75) Humidity and Temperature Sensor IC. [PDF-dokumentti].  
<[http://www.sensirion.com/fileadmin/user\\_upload/customers/sensirion/Dokumente/Humidity/Sensirion\\_Humidity\\_SHT7x\\_Datasheet\\_V5.pdf](http://www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/Humidity/Sensirion_Humidity_SHT7x_Datasheet_V5.pdf)>. (Luettu 18.7.2014)
27. Xtrinsic MPL3115A2 I2C Precision Altimeter. [PDF-dokumentti].  
<[http://cache.freescale.com/files/sensors/doc/data\\_sheet/MPL3115A2.pdf?pspl=1&Parent\\_nodeId=1307914182184706466838&Parent\\_pageType=product](http://cache.freescale.com/files/sensors/doc/data_sheet/MPL3115A2.pdf?pspl=1&Parent_nodeId=1307914182184706466838&Parent_pageType=product)>. (Luettu 19.7.2014)

